electronics

MDPI

Review

A Systematic Guide for Predicting Remaining Useful Life with Machine Learning

Tarek Berghout [1] and Mohamed Benbouzid [2,3,*]

[1] Laboratory of Automation and Manufacturing Engineering, University of Batna 2, Batna 05000, Algeria; t.berghout@univ-batna2.dz
[2] Institut de Recherche Dupuy de Lôme (UMR CNRS 6027), University of Brest, 29238 Brest, France
[3] Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China
* Correspondence: mohamed.benbouzid@univ-brest.fr

Abstract: Prognosis and health management (PHM) are mandatory tasks for real-time monitoring of damage propagation and aging of operating systems during working conditions. More definitely, PHM simplifies conditional maintenance planning by assessing the actual state of health (SoH) through the level of aging indicators. In fact, an accurate estimate of SoH helps determine remaining useful life (RUL), which is the period between the present and the end of a system's useful life. Traditional residue-based modeling approaches that rely on the interpretation of appropriate physical laws to simulate operating behaviors fail as the complexity of systems increases. Therefore, machine learning (ML) becomes an unquestionable alternative that employs the behavior of historical data to mimic a large number of SoHs under varying working conditions. In this context, the objective of this paper is twofold. First, to provide an overview of recent developments of RUL prediction while reviewing recent ML tools used for RUL prediction in different critical systems. Second, and more importantly, to ensure that the RUL prediction process from data acquisition to model building and evaluation is straightforward. This paper also provides step-by-step guidelines to help determine the appropriate solution for any specific type of driven data. This guide is followed by a classification of different types of ML tools to cover all the discussed cases. Ultimately, this review-based study uses these guidelines to determine learning model limitations, reconstruction challenges, and future prospects.

Keywords: damage propagation; deep learning; degradation analysis; machine learning; prognosis and health management; remaining useful life

check for updates

Citation: Berghout, T.; Benbouzid, M. A Systematic Guide for Predicting Remaining Useful Life with Machine Learning. *Electronics* **2022**, *11*, 1125. https://doi.org/10.3390/electronics11071125

Academic Editor: Wenxian Yang

Received: 8 March 2022
Accepted: 30 March 2022
Published: 1 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

RUL is an important real-time performance indicator of operating systems under working conditions. Indeed, RUL helps in providing necessary planning for condition-based maintenance tasks of such systems with an attempt to approach zero downtimes [1,2]. An online lifetime estimate can usually be performed following one of three paths: a physics-based model, a data-driven model, or a hybrid model of both [3,4]. Cutting-edge technologies in the industrial sector make systems' complexity surprisingly increase [5]. This subsequently leads to making physical modeling fail to provide useful simulation due to the vastness and dynamic behavior resulting from systems' higher level of flexibility [6,7]. In addition, the massive volume of data traffic makes it challenging to analyze using predictive residue-based models that are traditionally poorly generalized [8,9]. As a result, modeling standards are pushed further towards using data to mimic such complex behavior. Accordingly, model reconstruction based on ML has emerged and continues to advance in adapting to several cases of data complexity by targeting its attributes, i.e., volume, velocity, and variety (3 V) [10].

In the field of PHM, RUL prediction based on ML modeling has given rise to numerous studies. As a result, many comprehensive reviews have been devoted to studying these approaches, addressing different aspects related to classifications and learning paradigms. In this review-based study, and in an attempt to appropriately analyze recent and relevant studies to draw useful guidelines and suggestions, a structured research methodology is adopted. The research targets recent publications (i.e., review and research papers) in well-known databases published in the last five years, from 2017 to 2021. A list of specific keywords belonging to lexical sets of PHM is carefully selected, e.g., RUL, PHM, ML, and deep learning (DL). Besides, ML known methods are classified ranging from conventional, evolutionary computation to DL tools. Different learning paradigms such as reinforcement learning (RL) and transfer learning (TL) are given special attention, in addition to advanced generative adversarial networks (GANs) and graph neural networks (GNNs).

At first glance, a set of reviews are analyzed in chronological order in terms of ML investigations. For instance, in [11], authors typically approach PHM from different angles, including trends, issues, and technologies. In this context, they scrutinize "data-driven" approaches, which are considered to be the chosen approaches for PHM (see [11], §4.3). Authors in [12] provide a general overview of assessment methods used for SoH and lifespan of Li-ion batteries. ML has been introduced both as adaptive learning, i.e., Kalman filter, particle filter, and least squares, and as data-driven approaches, i.e., fuzzy logic, artificial neural network (ANN), and support vector machine (SVM). An interesting study is carried out in [13], where authors studied the use of DL tools, in particular, autoencoders (AEs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) in PHM; their review offers valuable perspectives for future works. In [14], authors tackled an important in-depth study on RUL prediction for machines. In their study, many important issues such as data acquisitions, predictive models, health index (HI), and health stages (HS) were discussed. In the ML section (see [14], §5), statistical and non-statistical approaches have been undertaken where the most important tools are listed with some examples from the literature. In [15], authors also focus on DL techniques used in PHM. More specifically, they studied different classes of DL such as AEs, deep belief networks (DBNs), RNNs, and CNNs. They also target used techniques for extracting data features in different domains including time, frequency, and time-frequency domains. Authors in [16] investigated the use of data-driven methods for assessing Li-ion batteries' SoH. They mainly discussed two different topics: SoH estimation (i.e., diagnosis) and RUL prediction (i.e., prognosis). In terms of diagnosis, they categorized ML models into three groups, namely, fitted characteristics of the model, processed external characteristics, and direct external characteristics with respect to various input characteristics driven by the training model. Concerning prognosis, they introduced ML consisting of two main types of learning models, namely, probabilistic and non-probabilistic methods. In [17], authors elaborated on a specific study on bearing RUL prediction. Particular attention was paid to common DL techniques in their review. In terms of used tools, AEs, DBNs, CNNs, and RNNs were studied. Additionally, GANs and TL are discussed in their review. The review introduced in [18] brought up different types of used methods in degradation modeling for RUL prediction. They discussed their use in traffic and transport related to wooden and concrete bridges. ML techniques are referred to as artificial intelligence (AI) methods, which more generally list all types of intelligent methods including scalable computational techniques. In [19], different approaches used for RUL prediction are introduced. Distinctively, data-driven methods are referred to as virtual models, where stochastic methods and DL are discussed. Stochastic methods include winner, gamma, and hidden Markov models (HMMs) processes, while DL includes DNNs, RNNs, and CNNs. Another study is introduced in [20], where authors discussed DL-based RUL prediction of Li-ion batteries.

Table 1 summarizes the above-discussed reviews in terms of the context in which ML is described in detail. From these review papers, while addressing among other interesting topics such as Big Data and the Internet of things (IoT) eras where data are massive and dynamic using DL, some interesting conclusions are drawn regarding the usefulness of ML for PHM. In addition, special attention is given to feature mappings (i.e., extraction details) that might improve mismatch in data distribution and model generalization.

Table 1. Most studied types of ML models in the literature.

Reference	Studied Types of ML Models
[11]	Data-driven in general
[12]	Adaptive learning and data-driven in general
[13]	**DL** models
[14]	Statistical and nonstatistical approaches
[15]	**DL** techniques
[16]	Probabilistic and nonprobabilistic methods
[17]	**DL**, GAN, and TL
[18]	Scalable computational techniques
[19]	**DL** and stochastic methods
[20]	**DL** models
[11]	Data-driven in general

Generally speaking, most of these studies manage to delve into citing well-known methods with detailed classifications. Meanwhile, in the context of "*providing guidance*" for predicting RUL or SoH, there is something of a scarcity in the general explanation of the followed methodology. Specifically, these studies focus on "*what has been done*" in RUL prediction and imperceptibly ignore "*how and why it was done in this particular way*", and it seems there are no specific details pointing to if "*this is the only way or not*" to predict RUL for a specific type of system or data. In this context, the important question to be answered should be: "*What is the specific type of problem for which we should use a specific class of ML models?*" In order to provide a comprehensive answer to this question and to overcome this gap of providing such important details, our contributions can be enumerated as follows:

1. To answer the raised question and provide guidance for readers and ML developers interested in RUL model reconstruction, an overall solution of any RUL prediction problem in a kind of flowchart that simplifies the selection of the appropriate ML modeling process is introduced;
2. After model selection, the training methodology instructions are discussed in depth for further explanation;
3. To ascertain the reliability of the proposed methodology, the proposed flowchart is justified by some of the most important examples of the recent literature that perfectly match the different cases of RUL prediction;
4. To discriminate the different classes of ML models used for RUL prediction, a detailed classification of ML models with the help of the proposed flowchart is thoroughly discussed;
5. By adopting the proposed flowchart, model reconstruction is made clearer and easier to be drawn;
6. A discussion of advantages, disadvantages, and limitations of some important ML tools from each class is also provided;
7. To remedy RUL prediction problems, prospective solutions are proposed.

This paper is organized as follows: Section 2 is devoted to describing RUL model selection guidelines. Section 3 discusses learning model training methodologies. In Section 4, a detailed classification of ML models according to several aspects, i.e., data availability, complexity, drift, and model complexity is provided. Section 5 provides important discussions and describes encountered challenges and limitations when designing RUL models. Section 6 is thereafter devoted to future improvements and opportunities.

2. RUL Model Selection Steps

According to Figure 1, there are three necessary steps that have to be followed to build an RUL prediction model, in particular, model selection, reconstruction, and prediction. Model selection is the most important step that was not significantly addressed in most of the above-analyzed works. In this context, the important question to be answered would be: *"How exactly do we choose our training models and what criteria should we use to do so?"* Accordingly, this section is specifically devoted to answering this question. It is convenient that the goal of a proper selection and reconstruction methodology is to accurately predict the RUL of new unseen samples. The term unseen samples in this case refers to new driven samples that have never been tested on the model before, in which the actual SoH estimate is completely dependent on them.

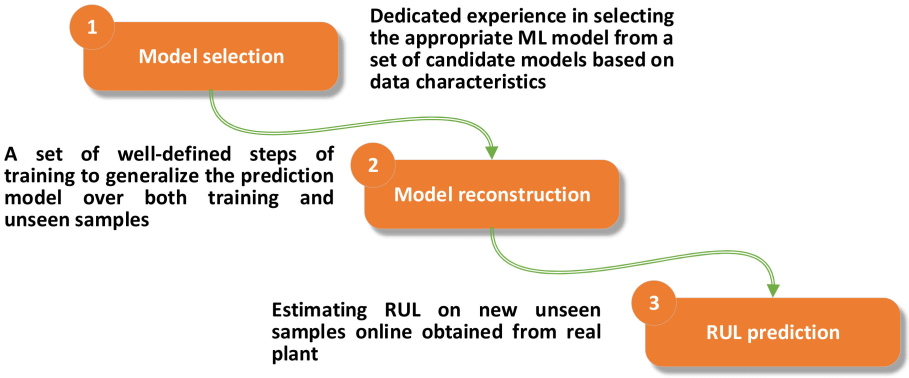

1
Model selection
Dedicated experience in selecting the appropriate ML model from a set of candidate models based on data characteristics
A set of well-defined steps of training to generalize the prediction model over both training and unseen samples
2
Model reconstruction
Estimating RUL on new unseen samples online obtained from real plant
3
RUL prediction

Figure 1. Necessary steps for constructing an RUL prediction model with ML tools.

2.1. Model Selection Guidelines

As illustrated by the proposed flowchart of Figure 2, model selection involves examining four main criteria, in particular, data availability, data complexity, data drift, and model complexity. This sub-section is dedicated to describing these selection criteria.

2.1.1. Data Availability

Data availability implies that training inputs and labels are available and complete. This completeness is closely related to the existence of the whole important run-to-failure measurements from the beginning of the life of the system until complete failure. Thus, a real-time recording process is required for the RUL labeling task. In this context, if data is complete, a direct RUL prediction by mapping inputs to the targets by solving a regression problem is the best solution. However, a further test on data complexity is necessary to determine the appropriate learning paradigms and whether to use conventional ML or DL. Contrariwise, if labels are missing, an HI and HS should be built to assess systems' SoH and RUL. Furthermore, if samples are missing such as in accelerated life tests, generative models (GMs) are required to remedy this incompleteness in the life path. Apart from GMs, domain adaptation (DA) methods such as transfer learning (TL) can also be used separately from or jointly with GMs to gain expertise from other domains (i.e., data fusion, label propagation, across modalities, etc.) and to help improve data discrepancy and generalizability of the model [21].

2.1.2. Data Complexity

Data complexity refers to the number of samples relative to their dimensions and dynamics, as previously stated as data 3V. In this context, types of used sensors measuring the industrial process conditions have a strong effect on the model selection process. Indeed, depending on the prognosis paradigm which targets external or internal degradation failures, images sensors (e.g., electroluminescence, thermographic, infrared, X-ray, etc.) and standard sensors (e.g., vibration, temperature, irradiation, etc.) can be exploited. Nonlinearity and recording measurements dimensions therefore indicate which model is needed to accomplish the approximation process [22]. Accordingly, the higher the 3V, the more complex the system is. In general, if data is massive and subject to a higher level of cardinality, then the constructed model should have the ability to learn from representations such as in DL models, otherwise, conventional ML is enough for universal approximation and generalization. Additionally, in this particular situation, a primary test on the collected samples by involving several ML models from both DL and conventional ML can give an insight into its complexity.

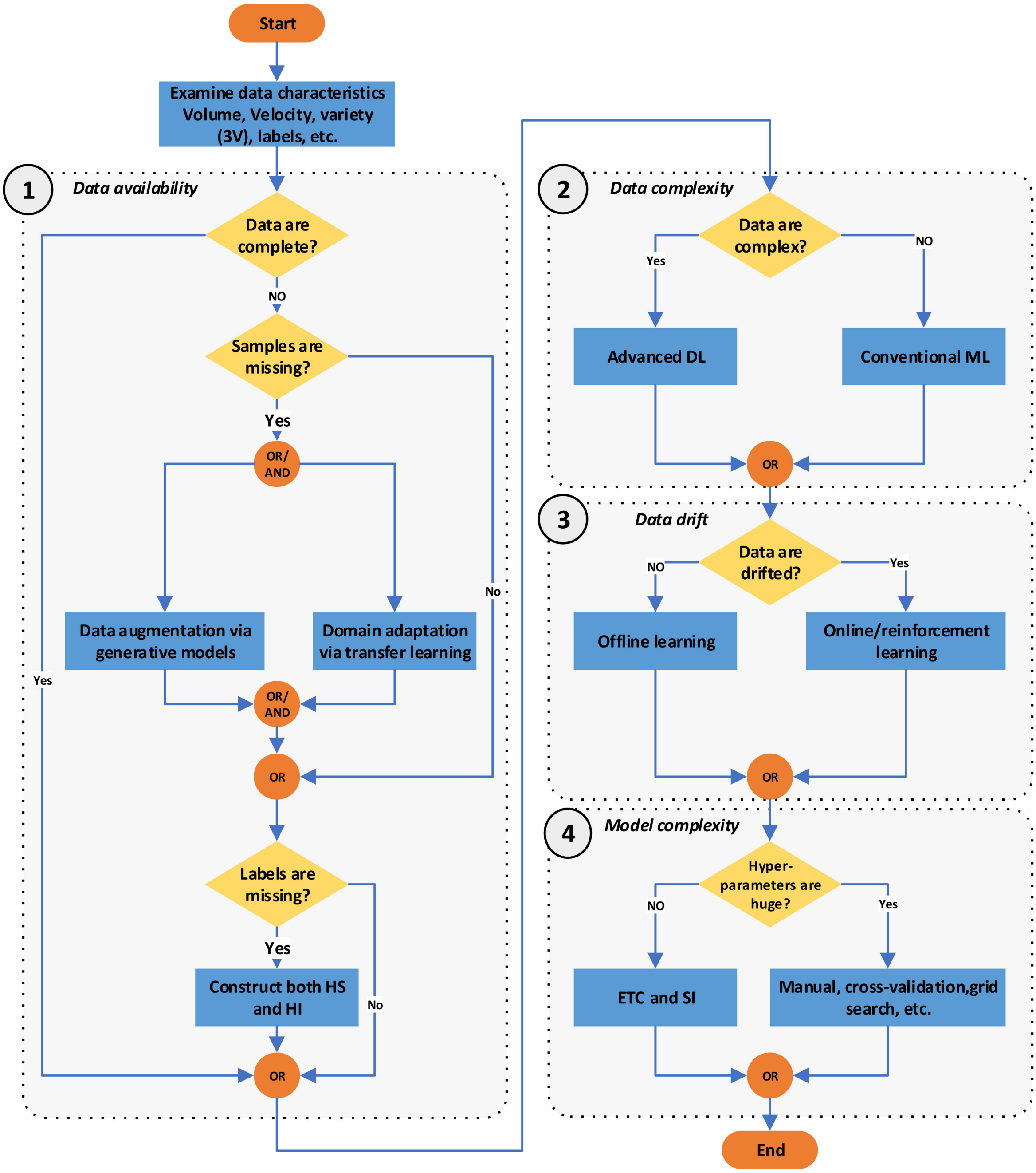


Figure 2. ML model selection methodology for RUL prediction.

2.1.3. Data Drift

Data drift is a concept used to describe continuous changes and dynamism in data at the time of its delivery process (i.e., run-to-failure samples). Ignoring this point definitely leads to performance degradation of the entire training as well as the model update process. Adaptive online learning models help to address these model performance degradation issues more than static modeling procedures by providing dynamic online updates with specific forgetting mechanisms (i.e., weighting) controlling generalizability and divergence of learning behavior of the ML model [23]. In PHM, the data drift phenomenon is the result of continuous change in working conditions as well as model SoH. In this context, two types of data can be distinguished: sequential time-series with a higher level of 3V and static offline data. Sequential data is a concept indicating that specific samples in a chunk of data depend on other points from other chunks in a sort of intercorrelation with respect to their order. Accordingly, static data indicates that it does not change after being recorded. After checking data types, the model is eventually selected with the help of the proposed flowchart (Figure 2). However, there is yet another issue related to the complexity of the learning model.

2.1.4. Model Complexity

Modeling complexity is generally related to model architecture that specifically depends on a set of learning parameters (e.g., weights, biases, and hyperparameters) to form the whole approximation function. It is also related to the number of involved hyperparameters. Accordingly, if the training process is expected to have a large number of hyperparameters (e.g., CNN with multiple mappings and adaptive convolutional filters), then the model is expected to be complex. The only possible solution, therefore, is to update hyperparameters, either with a grid search, which is computationally expensive, or by using a traditional exhaustive manual tuning by human intervention. Accordingly, the two methods are not quite accurate to determine or at least to approach optimal solutions. Alternatively, if the model only has a few sets of hyperparameters, evolutionary computation techniques (ECT) and swarm intelligence (SI) can be used to automate the learning process and optimize parameters selection. Specifically, one cannot guess exactly at what threshold of parameter number we can judge the complexity of the model. However, a routine primary test on a set of ML models (i.e., both DL and conventional ML) on specific hardware indicates the recommended path.

In an attempt to shed more light on these types of data, some examples are selected. Two main cases are considered, where the first example is tightly related to the well-known Commercial Modular Air-Propulsion System (C-MAPSS) simulation dataset, representing a prime example of data completeness, complexity, and drift [24]. In the meantime, the second example is specifically dedicated to discussing the incomplete data case, which addresses both data complexity and drift as well. Therefore, the PRONOSTIA bearings dataset is selected [25]. It should be mentioned that this review-based study is not so much about solving RUL problems as it is about giving guidance to solve the problem. Therefore, the reason for choosing only two datasets and not more or less is to explain these specific cases only. Moreover, in the following sections we will be able to conclude that the C-MAPSS and PRONOSTIA datasets are among the most used ones in the literature.

2.2. Complete Data

C-MAPSS data is the result of a simulation model of a specific type of turbofan engine, i.e., a two-spool engine with thrust power up to 400340N (Figure 3) [24]. In this model-based simulation process, run-to-failure measurements are recorded under different conditions. During simulation, data are also considered to be contaminated with noise from different sources to mimic real scenarios. As a result, the massive data are divided into 4 fault detection (FD) subsets (i.e., FD001, FD002, FD003, and FD004), where each subset contains a considered number of the engine life cycles.

We fully understand from this brief description that data are massive, while noise complexifies these data. Besides, the continuous change in working conditions leads to more dynamicity in data. Additionally, data are recorded in time series in an online sequence-by-sequence process. In this context, since data are available, complex, dynamic, and online driven, following the Figure 2 proposed flowchart, the most appropriate learning methodology should be an online adaptive DL model able to be dynamically updated in order to deal with time-series analysis.

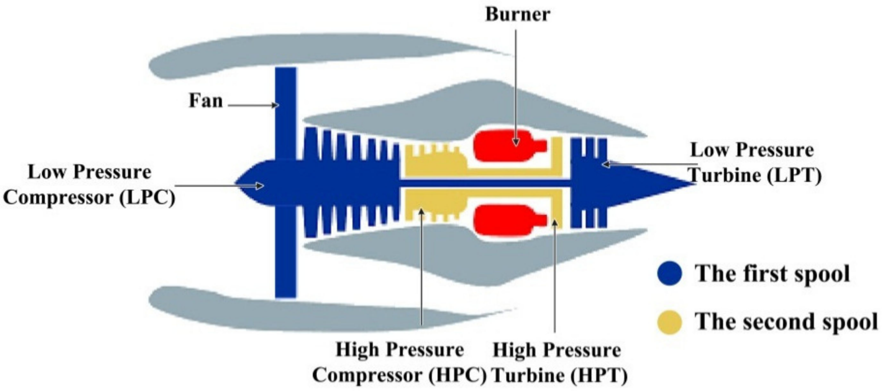

Burner
Fan
Low Pressure Compressor (LPC)
Low Pressure Turbine (LPT)
● The first spool
● The second spool
High Pressure Compressor (HPC)
High Pressure Turbine (HPT)

Figure 3. Studied turbofan type by C-MAPSS model.

In this context, Table 2 illustrates C-MAPSS and RUL examples from the literature on learning models. It is noticeable that most of the used training models are DL ones taking into account data dynamic change. Therefore, it is clearly shown that most of the methods are concerned with adaptive learning rather than ordinary offline learning. For instance, shown in Table 2, works proposed in [2,26–31] generally use RNN variants such as long short-term memory (LSTM), gated recurrent unite (GRU), and adaptive denoising online sequential extreme learning machine (OSELM). In the meantime, only a few studies do not consider adaptive learning such as in [32,33], where CNN is the main learning algorithm.

Table 2. Learning models used for complete and complex data.

Reference	Studied Types of ML Models
[26]	LSTM
[27]	LSTM
[32]	CNN
[28]	CNN and LSTM
[29]	LSTM
[33]	CNN
[34]	OSELM
[35]	GRU
[31]	LSTM

For illustration only and in an attempt to show some examples of health indicators (i.e., RUL is available in this case), a single life cycle from the first subset FD001 is chosen to highlight both RUL and data behavior. Figure 4 is provided to visualize data being studied in this particular life cycle. Figure 4a represents various sensor measurements collected over the entire life cycle of the engine, from its operation beginning until its complete failure. This is the reason behind the progressive deterioration of these measurements. Meanwhile, Figure 4b describes the desired RUL, which refers to the aging level by the real-life schedule. It shows that the RUL target function reflects the Figure 4a degradation process in a sort of linearly bounded function. This function was proposed in the 2008 PHM data challenge conference [36]. The reason behind this representation is that the engine is considered to be working under healthy conditions (i.e., stable phase in RUL function), and at a certain level, it starts to progressively deteriorate due to damage propagation in specific components (i.e., linear deterioration part in RUL function). Accordingly, the data-driven model mission, in this case, is to achieve the best approximation (i.e., curve fit) while dealing with all lifecycles by reducing the amount of both late and early predictions. Early prediction refers to the ML models suggested taking the necessary maintenance measures at an early stage. In fact, this is very important specifically when the model is about to avoid a detrimental situation of damaging the system. However, too early predictions result in higher maintenance resource consumption and financial losses. On the contrary, late predictions are very detrimental, which in real-world applications could result in catastrophic loss of equipment as well as loss of life due to a maintenance program scheduled at a later date. Figure 4c is a simple example that showcases encountered problems when predicting RUL with a linear regression model. Figure 4c is obtained by training an approximation model based on ordinary least-square estimation using sensors measurements as inputs and the linear bounded RUL function as a target. After that, we used the same input to be able to observe the training quality. The curve fit result labeled with early and later predictions shows that the model is driven by data towards late predictions (i.e., many predictions are in the late part). This type of prediction in such a case can be considered harmful to the engine due to possible delays in conditional maintenance planning.

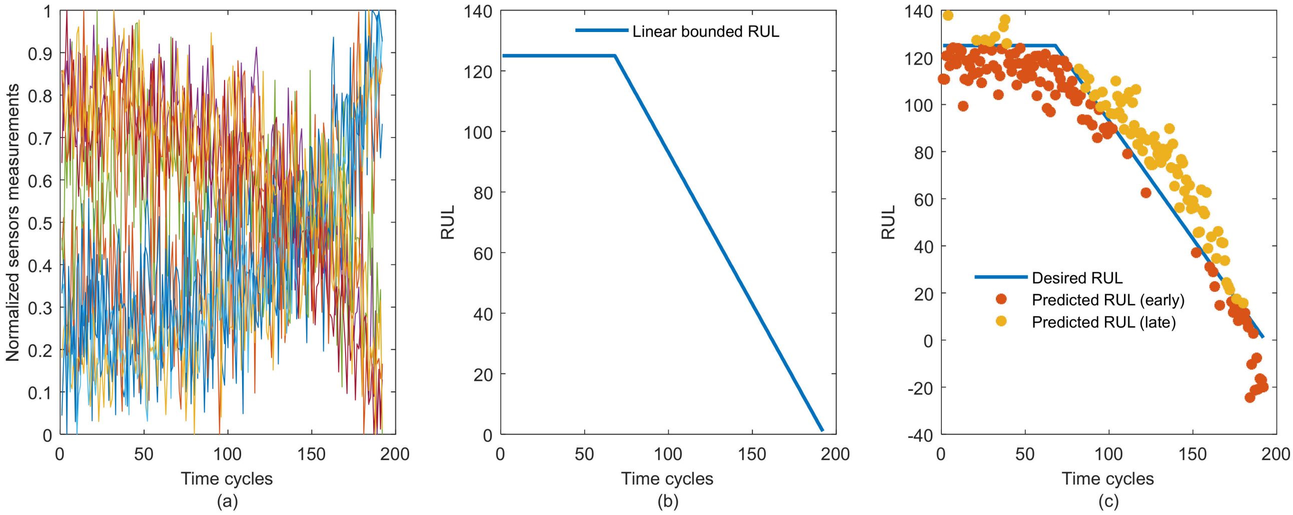


Figure 4. Illustration of data behavior in a single life cycle (i.e., cycle1 from FD001) from C-MAPSS dataset: (**a**) Run-to-failure sensors measurement during gradual degradation; (**b**) Defined RUL targets for life cycle; (**c**) Early and late RUL predictions resulted from the target curve fitting with a linear model.

2.3. Incomplete Data

As addressed in the introductory publication of the 2012 PHM data challenge in [25], PRONOSTIA is an accelerated life test platform designed to study ball bearings degradation. Data, in this case, are a perfect example, which fits the conditions of incomplete data with both missing degradation patterns (input samples) and labels. The PRONOSTIA dataset is recoded under different load and speed conditions. Run-to-failure measurements (i.e., temperature and vibration) are recoded with temperature and accelerometers sensors placed in different positions of the seventeen tested ball bearings as elucidated by Figure 5. The accelerated degradation process is used as an alternative to easily collect degradation patterns similar to real ones. However, since the conditions are not real (i.e., accelerated aging more specifically), learning patterns are subject to some loss of information. Besides, real RUL timing is no longer available due to the noncompatibility of life acceleration with real degradation cases.

In this case, also by projecting data characteristics onto the proposed flowchart (Figure 2), specifically the data availability part, 4 main processes must be accounted for, i.e., data augmentation and/or domain adaptation (DA), HI, and HS reconstruction. Data augmentation can be performed via GMs to improve and generate new examples to extend representations meanings. DA is also useful in reducing data distribution mismatch as well as obtaining additional generalization capability to improve model expertise. Accordingly, learning paradigms such as GANs and TL are very helpful. For example, in [37–39], TL is used to transform knowledge either through learning models or through the working conditions of different bearing life cycles. In [40], generative adversarial models were employed to extend data representation when predicting RUL. HI is a probabilistic function or performance indicator designed either from the input signals themselves (i.e., information fusion) or it could be either a linear or exponential degradation function [41,42]. HI is generally detected by solving a supervised trained approximation function [43,44]. In the meantime, HS indicates to which phase the operating behavior of the systems belong (e.g., operating normally, degrading, and complete failure). Generally speaking, HS can be determined either via signal processing tools or by solving an ML clustering problem [37,43]. However, unlike ML tools that could divide a single life path into several stages, SP techniques solve a single threshold division problem. This division is particularly known as the first predicting threshold (FPT), which is tightly related to a degradation first-time appearance. Table 3 summarizes the used ML tools in this particular case to solve the PRONOSTIA prediction problem.

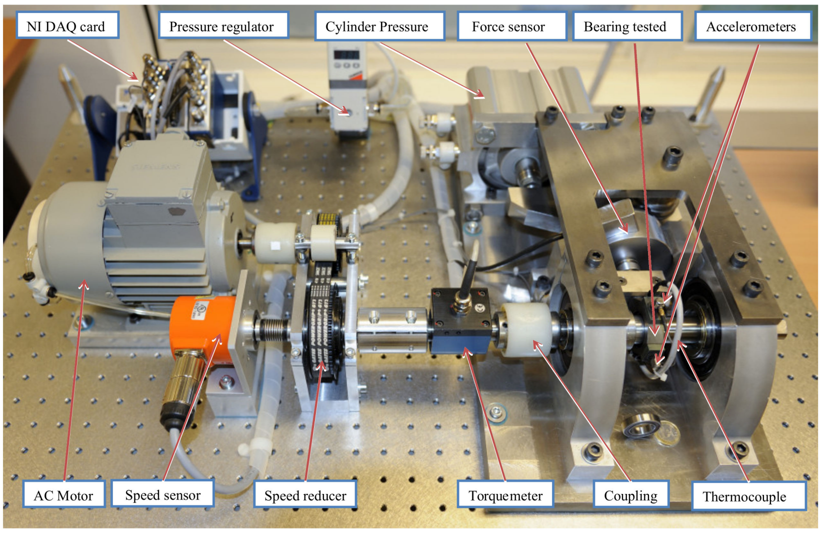


Figure 5. Illustration of the PRONOSTIA platform [25].

Table 3. Learning models used for incomplete unlabeled data.

Reference	Used ML Models
[41]	LSTM
[42]	RNN
[45]	Nonlinear stochastic model
[46]	Thresholding algorithms
[43]	AEs and MLP
[44]	Recursive filtering
[39]	TL, MLPs, and HMM
[40]	GANs and DL
[37]	TL, LSTM, and GMM
[38]	TL and DL

Figure 6 is an example that elucidates both HI and HS prediction in a single life cycle from the PRONOSTIA dataset. This bearing life cycle is extracted from run-to-failure vibration measurements of the first tested bearings (Bearing1-1). In this case, the main problem is the missing of some important samples and labels due to the accelerated aging process of PRONOSTIA experiments. Accordingly, we constructed both HI and HS to estimate SoH reflecting the aging level. As a result, we used the same linear regression model as in the Figure 4 experiment to illustrate HI prediction, while the Gaussian mixture model (GMM) cluster is used to assess HS, and HI is identified as an exponential deteriorated function as shown in (1).

$$HI(t) = d - e^{\lambda t + b} \tag{1}$$

where b is the convergence rate, and d and λ can be analytically calculated with respect to initial conditions of time-instant t, i.e., $HI(t_0) = 1$ and $HI(t_{end}) = 0$. It also shows an example of HI prediction with a linear regression model. The reason to refer to the degradation process by an exponential function in this situation is due to the acceleration in life, which exponentially drives bearings towards failure.

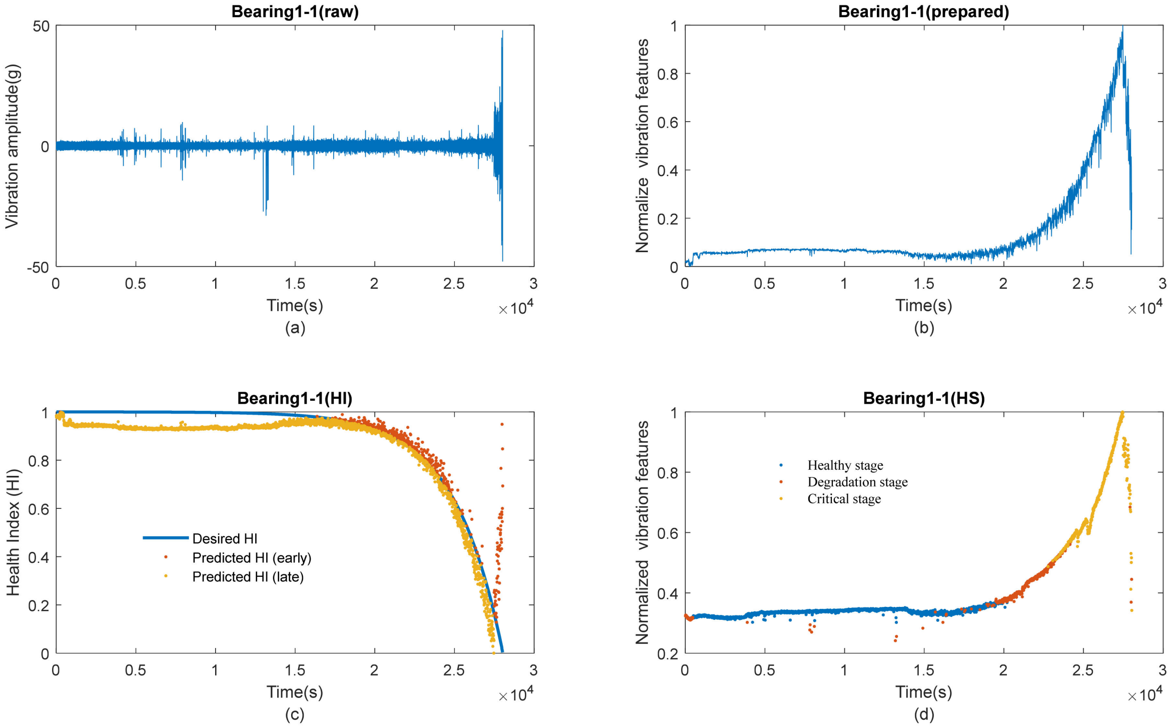


Figure 6. Example of health indicators in a single life cycle from PRONOSTIA dataset (i.e., "Bearing1-1"): (**a**) Raw run-to-failure vibration signal; (**b**) Prepared version of the signal; (**c**) Example of HI identification and curve fitting with a linear model; (**d**) HS divisions with GMM model.

Unlike the PHM 2008 dataset, and as we observe in raw signals as well as in the prepared signals from Figure 6a,b, respectively, data obtained from vibration signals present a nonlinear and nonstationary process. As a result, ML model reconstruction is subjected to a higher level of cardinality where samples with similar representations have different targets. These differences in responses between entities perturb the learning model by pushing it to wrong decisions. Besides, the lack of samples due to acceleration of life limits the model "*explainability*" so that it makes less sense in real-world applications. On the other hand, HI results in Figure 6c clearly indicate that prediction is happening at an early stage. This leads to increased maintenance costs due to early planning. HS division with a clustering process in Figure 6d helps in distinguishing between different bearings' SoHs and to classify health condition levels. HS is an additional metric to HI, which helps to further address the reliability of the prediction process.

3. RUL Model Training

After the learning model selection, the next step is training and validation. A well-structured methodology should be followed when doing so. Figure 7 illustrates the necessary steps for ML-based training, including data preprocessing, training through parameters tuning, and prognosis model evaluation.

3.1. Data Processing

Preprocessing is a necessary step to ensure that collected samples are ready for training. The goal is to remove any incoherent representations and to select only the most important ones, while the selected features, hopefully, equally contribute to the prediction process. Preprocessing also could utilize dimensionality reduction such as compression, sparse coding, and nodes pruning to reduce computational costs such as memory usage. In addition, appropriate feature mappings are useful in order to reach a more suitable data distribution. In this case, common methods of data preprocessing can be grouped into two main categories, in particular, signal processing (SP) and ML processing techniques.

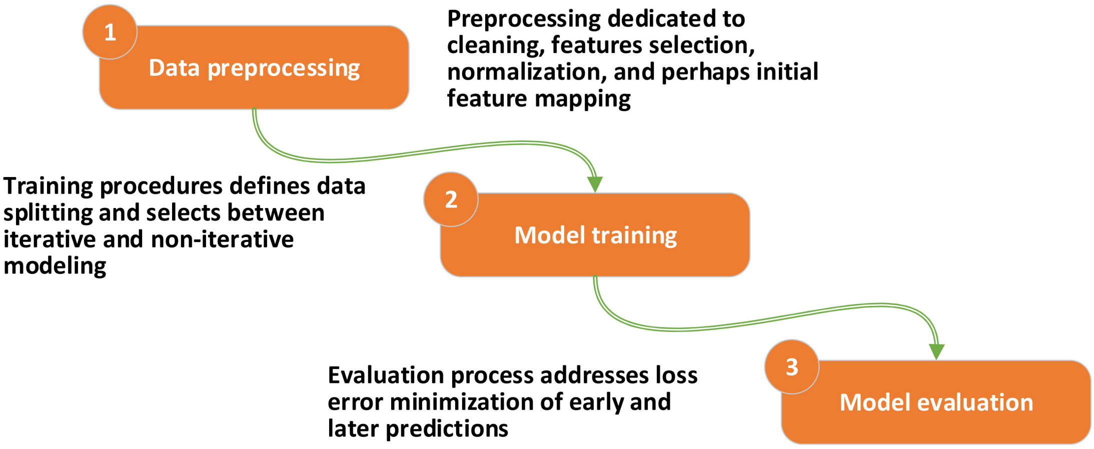

1
Data preprocessing
Preprocessing dedicated to cleaning, features selection, normalization, and perhaps initial feature mapping
Training procedures defines data splitting and selects between iterative and non-iterative modeling
2
Model training
Evaluation process addresses loss error minimization of early and later predictions
3
Model evaluation

Figure 7. Training steps of ML model for RUL prediction.

3.1.1. SP Preprocessing Techniques

SP techniques are generally nontrainable algorithms that follow certain algorithmic procedures to be able to handle good quality feature extraction. SP technique procedures are very important, especially when dealing with higher sampling rates and recording different signal types from multiple sensors. In the literature, SP techniques are well known in data preprocessing when feeding data-driven models, especially in the case of RUL prediction. For example, variational mode decomposition (VMD) is used within ML tools to improve acquired signals when recording media are noise-sensitive [47–49]. The Hilbert transform (HT) is also commonly used when extracting timely driven data mini-batches in a form of serially correlated samples, especially amplitude and frequency [50,51]. Similar to HT, the Hilbert–Huang transform (HHT) is more specifically used to treat nonlinear and nonstationary processes such as vibrations [52]. Power spectral density (PSD) is used to identify the amplitude in oscillatory signals, thus, it indicates at which frequency ranges variations are strong [50]. The Fourier transform (FT) is used to decompose a signal into its sine and cosine components. Thus, it is used in a wide range of applications, such as time-series analysis, filtration, reconstruction, and compression [53].

3.1.2. ML Preprocessing Techniques

Unlike SP techniques, ML preprocessing techniques are more automated learning algorithms and require less human intervention. Besides, ML preprocessing techniques, especially blackbox models, do not require strong background knowledge about signal processing. Among many ML preprocessing tools, the most important ones are mentioned in what follows. For instance, principal component analysis (PCA) generally depends on sparse representations, i.e., singular value decomposition (SVD) to be able to push feature representation into smaller and meaningful ones with fewer dimensions [54,55]. Compressed sensing (CS) is also a powerful data compression tool used in the field of prognosis. It is a kind of hybridization between sparse frequency domain and l_1 norm optimization, so it can be trained as any ordinary ML technique [56,57]. Additionally, AEs with different types (e.g., restricted Boltzmann machine (RBM), denoising AEs, variational AEs, convolutional AEs, and sparse AEs) are GMs used for reconstruction, compression, and extraction. The main purpose is to generate new samples in an unsupervised learning way to help improve the supervised learning model generalization during fine-tuning [2,58,59]. Preprocessing technique choice depends on the nature of data-driven samples. If the data type is a nonstationary time series that suffer from a higher level of nonlinearity, then SP techniques are more appropriate. Otherwise, a consistent ML preprocessing scheme is utilized for those not skilled in signal processing in most cases.

3.2. Training and Validation

In this section, while still considering the Figure 2 flowchart, we specifically delve into the description of the proper learning way by discussing the most important steps, including both data splitting and parameter tuning.

Generally speaking, if training and testing sets are already defined by experts in the field familiar with data quality and model fit, the training process should follow the same procedures to assess the behavior of the learning model during training or prediction on new unseen samples. Similar cases can be found in the previous cases of C-MAPPS and PRONOSTIA datasets where data are already split. However, an additional validation set that can be derived from a training data set to further judge the accuracy of the model is of great benefit. Alternatively, commonly used splitting techniques such as random sampling, bootstrap, Kennard–Stone, joint distances, and cross-validation algorithms can be based on the training process. However, the division process completely depends on the size of the used data as properly addressed in [60].

Another important case that must be discussed is the hyperparameters tuning procedures. Hyperparameters are the most important elements controlling loss function convergence conditions related to approximation. The above-described data selection methods can be used to select appropriate parameters from a randomly generated population grid [61], although grid search and ECT and SI can be adopted in this case to provide further optimization and shift closer towards global minima of the loss function [62]. After selecting the appropriate approach for hyperparameter optimizations, the learning process continues and the RUL model is built upon validation conditions, which are presented in the next subsection.

3.3. Evaluation

Evaluation of the RUL prognosis models is quite different from default learning procedures. As a result, the objective function to be minimized is not the same as a familiar loss function. Indeed, RUL curve fit is not similar to ordinary curves, where early and late predictions have a different impact on the maintenance decision process (i.e., not only reducing the distance between estimated and desired responses). Therefore, considering those types of predictions is crucial. In this context, different metrics were developed to describe the amount of variation in those prediction types and also to determine the prediction model accuracy. From a mathematical point of view, these metrics are usually formulated differently. However, they all agree on one and the same phenomenon that consists in penalizing late predictions, which are more damaging for the system than early ones.

Previous PHM challenges (2008 and 2012) can be considered to remedy this issue. In the 2008 PHM challenge, the score functions in (2) was used to assess the learning model accuracy, where N is the number of samples and $\{\widetilde{y}, y\}$ are predicted and desired RUL, respectively. As it is observed, late predictions are penalized with a penalization parameter equal to 13, while early ones are penalized with 10. These parameters are defined by experts' knowledge and also according to specific experimental conditions.

$$s = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} e^{\left(\frac{\widetilde{y}-y}{13}\right)} - 1, (\widetilde{y}-y) > 0 \\ \frac{1}{N}\sum_{i=1}^{N} e^{\left(\frac{\widetilde{y}-y}{10}\right)} - 1, (\widetilde{y}-y) \leq 0 \end{cases} \tag{2}$$

Figure 8 is the result of the application of Equation (2) on the curve fit previously shown in Figure 4c and showcased again in Figure 8a. It should be noted that the accuracy formula allows solving a minimization problem that attempts to reach a "zero" score. Therefore, distributions of its values far from "zero" entails that the model is not accurate, otherwise, the closer to "zero", the more accurate the model is. In this case, we are noticing early errors closer to zero while late errors are even far from zero. This means that the trained ML model is a late predictor more than an early one.

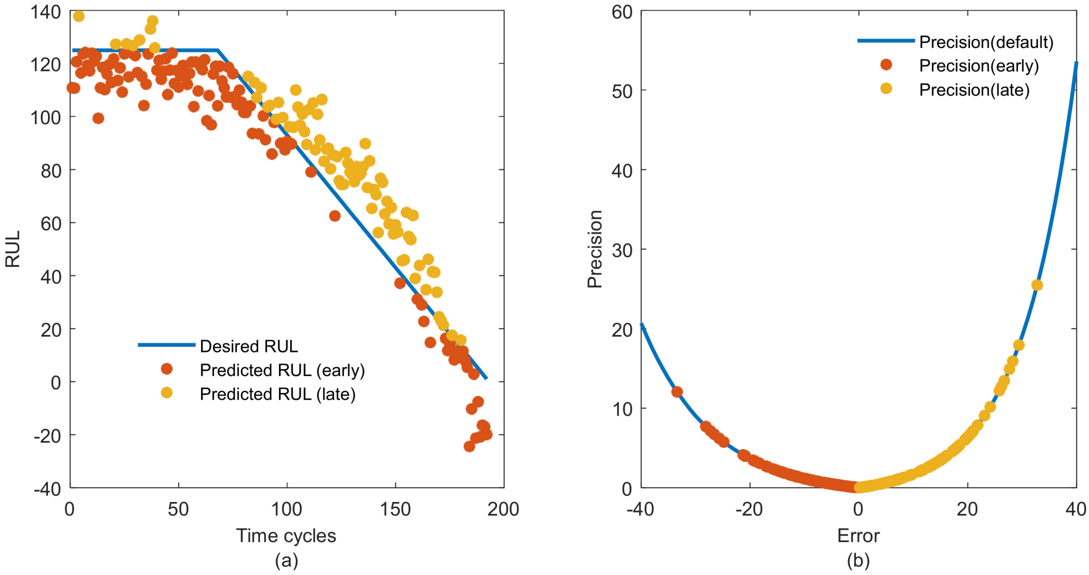


Figure 8. Applying PHM 2008 accuracy analysis formula on RUL curve fit: (**a**) RUL curve fit with ML linear model; (**b**) Predictions distributions according to PHM 2008 accuracy formula.

Another example related to ML models' evaluation under incomplete data can be given based on the 2012 PHM challenge. In this case, a similar scoring formula is used to evaluate HI prediction given in (3). Similarity remains in terms of penalization of early and late predictions, which always follows specific rules related to maintenance planning.

$$s = \left\{ \begin{array}{l} \sum\limits_{i=1}^{N} e^{\left(\frac{-\ln(0.5)E_i}{20}\right)}, E_i > 0 \\ \sum\limits_{i=1}^{N} e^{\left(\frac{-\ln(0.5)E_i}{5}\right)}, E_i \leq 0 \end{array} \right. \Bigg| E_{i=}(y_i - \widetilde{y}_i)/y_i \quad (3)$$

Unlike the PHM 2012 score function, this formula allows maximizing the objective function in an attempt to reach the value "one". Consequently, the closer the score value is to value "one", the more accurate the model. Figure 9 illustrates the application of this formula on the HI curve fit obtained in Figure 6c and showcased again in Figure 9a. By observing the accuracy results distribution, we can notice that early predictions are more largely heading towards value "one" than the late ones. In this context, this type of prediction is an early prediction that could lead to increased consumption of maintenance resources if prevention programs are incorrectly planned.

There is an important remark to consider when dealing with the "*explainability*" of the learning model. Indeed, accuracy metrics are used for optimization purposes and to indicate important information, but they have no real meaning in real-world applications. Therefore, an additional metric, such as the root mean squared error (RMSE) for instance, as in (4), is necessary to at least provide the amount of variation between predicted and desired responses. Besides, in this context, additive metrics such as mean absolute error (MAE), mean squared error (MSE), R^2, mean absolute percentage error (MAPE), etc., can also be exploited to further confirm results reliability.

$$e = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(\widetilde{y}_i - y_i)^2} \quad (4)$$

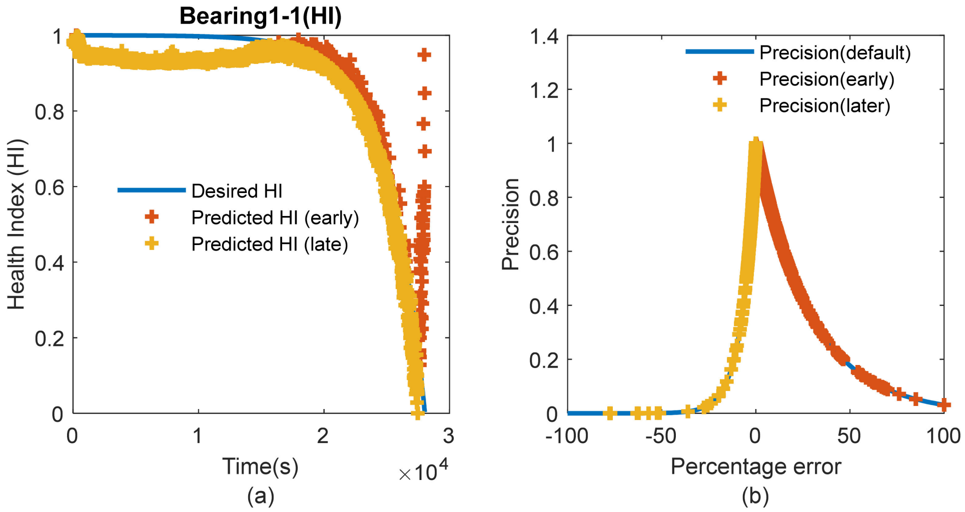


Figure 9. Applying PHM 2012 accuracy formula on HI curve fit: (**a**) HI curve fit with ML linear model; (**b**) Predictions distributions according to PHM 2012 accuracy formula.

Concerning HS splitting, and since ground truth labels are unavailable in this case, there is a scarcity in the evaluation of the clustering model capability. However, we can at least consider other metrics that hopefully indicate if such a cluster is capable of evaluating whether the data can be divided into a specific number of classes or not. These metrics have the ability to measure classes' dispersion when using specific clusters. For example, the Silhouette coefficient is used to evaluate clustering performances for the PHM 2012 dataset [37,63,64]. Formula (5) denotes the analytical expression for the Silhouette coefficient α, where ω and γ are the average and smallest distances between classes in the decision space, respectively [65].

$$\alpha = \frac{\omega - \gamma}{\max\{\omega, \gamma\}} \quad (5)$$

4. Classification of ML Models for RUL Prediction

Model selection, training, and evaluation are important steps to build an accurate training model. This section then provides a more detailed classification of ML methods of RUL prediction, which further helps in selecting learning models. A literature review is also proposed providing more details on data nature, solved problems, and chosen models.

According to the proposed classification shown in Figure 10, ML models for RUL prediction can be classified into one or a combination of six categories, including conventional ML, advanced DL, RL, ETC within SI methods, GM, and DA. In fact, this current classification does not particularly designate supervised and unsupervised learning models as subclasses of RUL models as in ML classification. This is because our goal is still to provide this particular RUL evaluation, which is generally supervised learning. Consequently, their description within learning algorithms when describing the literature methods is therefore considered.

4.1. Conventional ML

Conventional ML tools are used to train the prediction model for an approximation on a specific training set trying to obtain the best generalization on new unseen samples when trying to recognize patterns. As shown in Figure 11, the main objective is to approximate a specific set of inputs x to a specific set of targets y as they are presented. In fact, the use of such a feature extraction or any mapping function ϕ(x) is a nontrainable modeling process and a completely independent task of preprocessing.

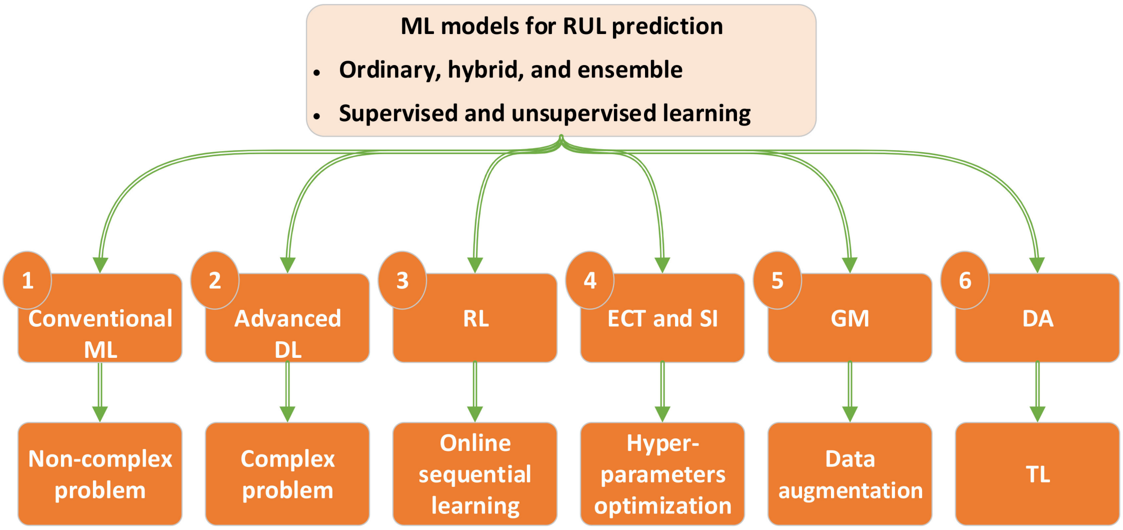

ML models for RUL prediction
Ordinary, hybrid, and ensemble
Supervised and unsupervised learning
1
Conventional ML
2
Advanced DL
3
RL
4
ECT and SI
5
GM
6
DA
Non-complex problem
Complex problem
Online sequential learning
Hyper-parameters optimization
Data augmentation
TL

Figure 10. Different classes of ML models for RUL prediction problems.

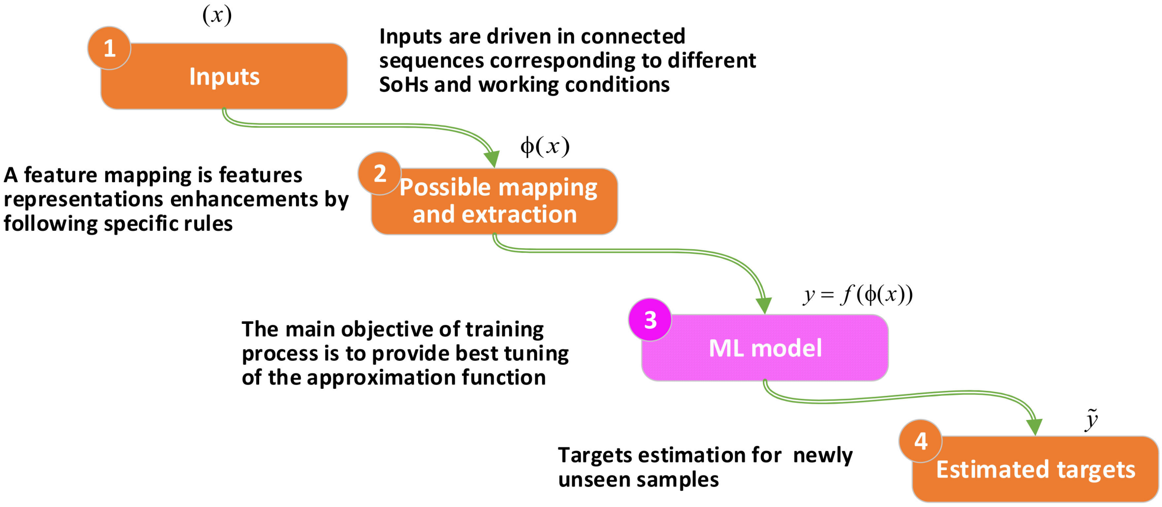

(x)
1
Inputs
Inputs are driven in connected sequences corresponding to different SoHs and working conditions
ϕ(x)
A feature mapping is features representations enhancements by following specific rules
2
Possible mapping and extraction
y = f(ϕ(x))
The main objective of training process is to provide best tuning of the approximation function
3
ML model
ỹ
Targets estimation for newly unseen samples
4
Estimated targets

Figure 11. Different steps to solve a supervised learning problem with conventional ML tools.

Models are generally shallow and depend on ordinary full rank mapping or kernels to be able to provide enough linearly independent representation able to minimize the loss function. Conventional ML can simply be presented as in (6), where $\tilde{y}$ are the estimated targets, ϕ(x) is the initial feature mapping and extraction process (independent from training), and f represents the designed ML model.

$$\tilde{y} = f(\phi(x)) \tag{6}$$

In this context, only well-known models were selected to showcase some examples. Indeed, methods such as support vector machines (SVM), multilayer perceptron (MLP), k-nearest neighbor (KNN), and ELM are thoroughly discussed.

4.1.1. SVM

SVM is a class of supervised learning algorithms interested in studying vector supports of specific data points to perform classification, regression, and outlier detection [66]. SVM was used in [67] to predict aircraft engines' RUL. It was subjected to modified similarity to be able to adapt with degradation analysis using the unlabeled PHM 2008 dataset. HI of degradation cycles was derived from deterioration paths themselves, leading to a more accurate approximation. The MAPE was used as the main metric for evaluation. In [68], authors used SVM for RUL prediction of Li-ion batteries. Both classification and regression characteristics of SVM are used in this study. A portion of discharging data (i.e., 70%) is divided into different HSs using specific classes defined by users. After that, these specific classes were used to train an SVM-based classifier for SoH estimation. Results of SoH classes are used to feed the SVM-based regression for RUL prediction. Classification accuracy, MAE, RMSE, and MSE were used as evaluation metrics of the ML model. Different from the previous works, authors in [69] combined SVM with an autoregressive integrated moving average model to early predict the RUL of aircraft engines ranging from 1 to 5 times unite. C-MAPSS dataset was used in this case, where R^2, RMSE, and MAE are the main decision criteria. In [70], an SVM classifier is used to train the HS splitting model using bearings' degradation cycles. Each life cycle is therefore split into five stages to divide necessary information of actual SoH. Accelerated life test datasets of PRONOSITA and intelligent maintenance system (IMS) bearings datasets were used to evaluate the proposal. Classification metrics are therefore used to evaluate the RUL model. In [71], a similar methodology of joint classification–regression approach is used to predict aircraft engines' RUL. Among many ML and DL methods, SVM was also discussed.

By comparing ordinary SVM with the multistage one, the results of the proposed (multistage) approach are clearly improved. In this context, and following simple voting decisions, better results can be achieved using SVM for SoH classification before feeding the HI or RUL prediction model [68,70,71].

4.1.2. KNN

KNN generally depends on unsupervised learning procedures where distances between classes in the neighborhood play an important role in decision making [72]. KNNs are also widely used in the PHM field. For instance, in the experiment introduced in [73], authors used the KNN approach along with a least-square one to train an ANN for RUL prediction of insulated gate bipolar transistors (IGBTs). In [74], a KNN regression model was used to estimate the RUL of Li-ion battery cells. Hyperparameters are therefore tuned with a differential evolution technique for optimal loss function minimization. Estimation and relative errors are the main criteria of model evaluation. In [75], authors elaborated a study on using KNN for HS classification of ball bearings. Time-domain features have been extracted from bearing life cycles that were recorded based on acoustic emissions.

According to the above-cited works, it seems that the best use of KNN in RUL prediction is for HS classification. This is due to the KNN design that allows performing data points similarity analysis.

4.1.3. MLPs

Multilayer perceptrons are shallow neural networks using nonlinear feature mapping with specific types of activation functions. Activations entail moving from one feature space to another one where meaningful representation provides more approximation [76]. MLPs are widely investigated in PHM as they achieved promising performances. For instance, in [77], HMMs were used to model a tool wear SoH and estimate its RUL. In this context, an MLP is used to help in estimating the observation probability. Transition probability of Markov chain and observation probability were used together to online estimate the RUL. In [78], a framework for RUL model reconstruction and parameters tuning was proposed. The model adopts ordinary MLPs within ECT for hyperparameters optimal search. The method is evaluated on a mechanical system specifically related to the previously discussed C-MAPSS dataset. In [39], the authors proposed and tested a data-driven approach on the PRONOSTIA dataset. HMM is used to locate the SoH change. After that, an MLP based on TL is used to solve data discrepancy problems. In [79], authors used vibration signals to estimate the RUL of the timing belt in an internal combustion engine. Accelerated life test experiments were carried out to determine fault threshold and acquire run-to-failure measurements, respectively. After a well-structured data preprocessing, MLPs were able to achieve acceptable prediction accuracy.

It is noticeable that MLPs are specifically exploited for direct predictions without considering adaptive learning under a wide range of dynamically changed data.

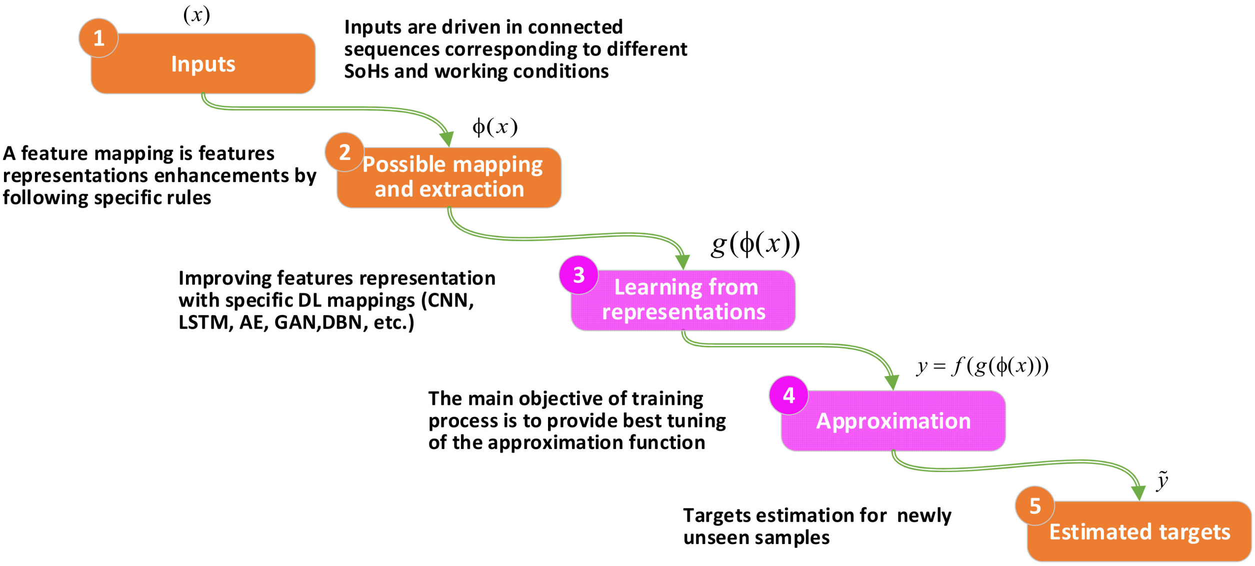
4.1.4. ELM

ELM is a very fast training method, which depends on least-square rules to train ANNs. Firstly, it was proposed to train a single hidden layer feedforward network. Then, it extends to fit any type of neural network learning architecture, including deep complex architectures [80]. Due to the ELM simplicity and higher accuracy, acceptance for PHM is also witnessed. For instance, in [1,2], ELM is used to train deep networks (i.e., a sort of DBN) for RUL prediction by using the C-MAPSS dataset. Algorithms were treated to be adaptive learners able to fit data changes in a sequential way. In [81], an enhanced OSELM architecture was proposed for the RUL prediction of integrated modular avionic systems. The neural network was therefore reinforced with a robust denoising AE to be able to learn efficient representations from data. A forgetting mechanism considering a forgetting factor was used to adapt the hidden layer generalization with data changes. In [82], a new type of loss function was given to the ELM to provide better regression robustness when predicting the RUL of aircraft engines. Accordingly, and similarly to MLPs, ELM also has been used for direct RUL predictions.

4.2. Advanced DL

DL is a subclass of ML that focuses on solving complex problems, typically related to big data environments [83,84]. Unlike conventional ML models, which are generally concerned with approximation, DL learning algorithms are more focused on representations than approximation (Figure 12). Thus, obtaining more meaningful feature space using well-defined layers of nonlinear abstractions leads to more universal approximation and generalization [85]. By following the same mathematical representation methodology, which was previously used in (5), the DL process could be presented as in (7), where g is a series of DL nonlinear mappings, such as convolutional mapping, autoencoding, recurrent mapping, etc.

$$\widetilde{y} = f(g(\phi(x))) \tag{7}$$



(x)
1
Inputs
Inputs are driven in connected sequences corresponding to different SoHs and working conditions
$\phi(x)$
A feature mapping is features representations enhancements by following specific rules
2
Possible mapping and extraction
$g(\phi(x))$
Improving features representation with specific DL mappings (CNN, LSTM, AE, GAN,DBN, etc.)
3
Learning from representations
$y = f(g(\phi(x)))$
The main objective of training process is to provide best tuning of the approximation function
4
Approximation
$\tilde{y}$
Targets estimation for newly unseen samples
5
Estimated targets

Figure 12. Different steps to solve a supervised learning problem with advanced DL tools.

Recently, DL has emerged in all application areas of ML, especially since the emergence of massive amount of data in the era of IoT and Industry 4.0. Supervised DL tools such as LSTM, CNN, DBN, and GNN have been widely investigated for RUL prediction, while unsupervised DL such as AEs are generally used as GMs for feature extraction. In this context, this section is dedicated to describing some of the major relevant works conducted using these tools in PHM.

4.2.1. LSTM

LSTM is a class of RNNs used to deal with sequential data, specifically time-series analysis. In other words, LSTM is an alternative to RNNs since they are unable to deal with the vanishing gradient problem caused by unrolling the hidden layer several times [86,87]. LSTM is the most appropriate tool to handle dynamic data such as in RUL problems. In [27], a vanilla LSTM, which is a type of LSTM variant widely used when dealing with language processing, was used to predict the RUL for aircraft engines using the C-MAPSS dataset. In [88], authors improved learning rules of LSTM to be able to more accurately predict the RUL of Li-ion batteries. The improvement targets the learning inputs, whereas LSTM generally uses a single input to match a single target. In this context, LSTM used many inputs to match a single target to provide further generalization. An interesting study has been investigated in [89], where a sort of hybridization between CNN and LSTM resulted in the construction of a new network called the convolutional LSTM. These hybrid representations allow both robust extraction of CNN while keeping powerful adaptive learning characteristics of LSTM at the same time when predicting bearings RUL.

In general, LSTM is a perfect learning algorithm for RUL prediction, specifically when dealing with big data that are sequentially correlated to each other.

4.2.2. CNN

CNNs are a class of artificial neural networks used for pattern recognition within higher-dimensional data. CNNs are widely known for their classification capability when dealing with images. Convolutional filters and pooling layers are perfect dimensionality reduction and feature extraction layers of CNNs [90]. In fact, a CNN helps in segmentation and patterns localization in a specific set of features belonging to a single sample. Generally, CNNs can be used without expertise in signal processing and raw data can be directly fed to the learner. In PHM field, CNNs are also well used for RUL predictors. In [91], a multiscale CNN is used for HI prediction. The PRONOSTIA dataset is utilized for comparison purposes. In [92], authors proposed a double CNN for the RUL prediction of bearings. In [93], a hybrid RNN–CNN algorithm was constructed to achieve both dynamic adaptation and approximation when estimating the HI of bearings life cycles obtained from the PRONOSTIA dataset.

In summary, CNNs are widely used for problems with big dimensions, either as a feature extractor or dimensionality reduction algorithm before approximation. The reason for adding recurrent units (i.e., RNN or LSTM) is to include an adaptive learning capability to be able to handle dynamically changing time-series data.

4.2.3. DBN

DBNs are a type of ANNs well known for their feature extraction capability. DBNs typically combine a stack of serially connected AEs of any type, before connecting a final layer for fine-tuning the approximation function [94]. It starts with the stacking of RBMs and then expands to accommodate all types of AEs. DBN is known for its applications in many fields of ML besides RUL prediction. In [95], a DBN was used under a big data environment to train the learning model to predict the RUL of rotating components. A set of stacked RBMs was trained by minimizing the contrastive divergence [96]. After that, the RUL model was fine-tuned for supervised learning. Authors in [97] used a stack of RBMs for unsupervised HI extraction from an aircraft engine degradation path. After that, a particle filter was used to tune the DBN for the RUL prediction. RBMs are trained with a contrastive divergence algorithm, while the particle filter is improved with a fuzzy inference system. In [98], a denoising algorithm based on DBNs and a self-organizing map was proposed to improve acquired signals from a wind turbine gearbox. After that, a particle filter was optimized by a fruit fly optimization algorithm for fine-tuning and supervised learning to reconstruct an RUL prediction model. Authors in [99] proposed a DBN optimized by a Bayesian approach and a hyper-band algorithm for the RUL prediction of supercapacitors.

It should be mentioned that DBNs are very powerful approximation tools specifically when data are significant and suffer from noise and higher cardinality.

4.2.4. Autoencoders

Autoencoders are a type of unsupervised learner able to achieve a higher level of accuracy when extracting meaningful representations [100]. AEs tasks differ from an application to another depending on data preprocessing requirements, including but not limited to denoising, compression, extraction, and neurons pruning. In [101], AEs were used for features compression when feeding a DNN for bearings RUL prediction. In [101], the AEs were used for sparse representations, which are a sort neuron pruning in a TL scheme. These AEs are trained to feed a supervised learning model for RUL prediction of a cutting tool. Authors in [102] combined a conditional variational AE with particle filter learning rules for RUL prediction of Li-ion batteries.

4.2.5. GNNs

GNNs are a type of deep ANNs designed to process data presented in the form of graphs. GNNs can be mined on graphs and provide an easy way to perform prediction tasks at node, edge, and graph levels [103]. GNNs have also been used in the PHM field. For instance, in [104], authors used a directed acyclic GNN model that combined CNN and LSTM networks for RUL prediction of aircraft engines. In [105], using GNNs' similar learning philosophy, a CNN was adopted to determine the RUL of aircraft engines using the C-MAPSS dataset.

An important feature of GNNs over an ordinary DL is that GNNs are able to capture the graphical structure of data, which are often very rich and difficult to achieve by an ordinary DL.

4.3. ECT and SI

ECT is a branch of SI algorithms that studies the development of bio-inspired algorithms derived from both natural evolution and biological systems [106]. Mathematically speaking, the common feature between these types of algorithms is that they are trained to optimize a randomly assigned initial population set to obtain the best individuals as a solution. The search mechanism and population updates are based on mathematical formulas inspired by real biological or swarm behaviors. In ML, these techniques are popular in hyperparameters optimization as they are able to elect very useful particles from initial random populations. They are also very helpful to address automatic learning better than other selection methods such as cross-validation or grid search and manual running. Figure 13 provides an overview of how ECI and SI are used within ML models. From a mathematical point of view, the optimization problem in ML can be simplified to fit the formula presented in (8), where l is the training model loss function (i.e., it could be any type of ML model).

$$f_{\min} = \text{Minimize}(l) s.t \|\widetilde{y} - y\| = 0 \tag{8}$$

In PHM, methods such as SI algorithms including particle swarm optimization (PSO), genetic algorithms (GAs), frog colonies, ant colonies, cuckoo search, and many other algorithms can be used. After a well-structured biographical search methodology targeting such a type of algorithms, we found that GA and PSO received special attention in constructing ML-based RUL prediction algorithms. In this context, this section is devoted to only investigating these algorithms.

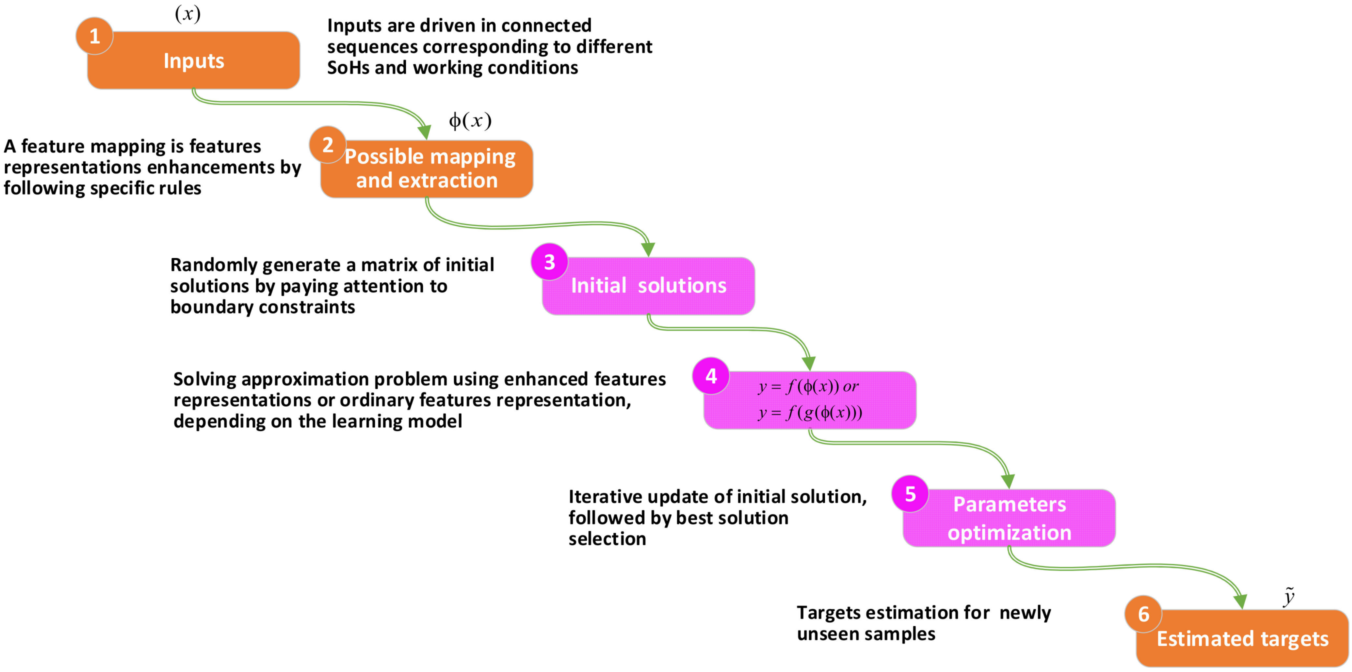

(x)
1
Inputs
Inputs are driven in connected sequences corresponding to different SoHs and working conditions
ϕ(x)
A feature mapping is features representations enhancements by following specific rules
2
Possible mapping and extraction
Randomly generate a matrix of initial solutions by paying attention to boundary constraints
3
Initial solutions
Solving approximation problem using enhanced features representations or ordinary features representation, depending on the learning model
4
y = f(ϕ(x)) or
y = f(g(ϕ(x)))
Iterative update of initial solution, followed by best solution selection
5
Parameters optimization
Targets estimation for newly unseen samples
ỹ
6
Estimated targets

Figure 13. Overview of using ECT and SI within ML models.

4.3.1. PSO

PSO is an SI computational method inspired by swarm behaviors designed to solve specific iterative optimization problems while trying to approach the necessary quality metrics [107]. In [108], PSO was used within a particle filter to predict the RUL of Li-ion batteries. This study proved that the PSO objective function could easily and deeply converge depending only on a small population. In [109], PSO was used to optimize SVM parameters for Li-ion batteries' RUL prediction. In [110], PSO was adopted to optimize LSTM parameters when analyzing journal bearing seizure degradations. In [111], authors proposed PSO itself for direct RUL prediction of Li-ion batteries. PSO can then be utilized for hyperparameters tuning or RUL prediction. However, according to the above-discussed works, using PSO for tuning hyperparameters is more beneficial.

4.3.2. GAs

GA is a heuristic ETC random search method inspired by natural evolution theories. GA reflects natural selection where chromosomes are elected for the reproduction of the next genetically improved generation [112]. Similar to PSO, GA is used for the same purposes of parameters selection of ML models when predicting RUL. In [113], a GA was used to tune the hyperparameters of a DL-based approach (i.e., RBM within LSTM) for the RUL prediction of aircraft engines. In [114], a GA was involved in an ensemble learning scheme of RUL prediction. Evaluation procedures were carried out using bearing datasets of IMS [115]. In [116], a GA was adopted to tune LSTM hyperparameters when finding an optimal local minimum to predict the RUL of supercapacitors. In [117], in another contribution related to the RUL perdition of Li-ion batteries, a GA was also used for hyperparameters tuning of an SVM algorithm. It is obvious that the optimal use of GAs lies in the tuning of hyperparameters following an accurate selection scheme.

4.4. RL

RL is one of the most interesting research topics in modern AI. It is a type of learning that allows an agent to learn in an interactive environment by trying and correcting the mistakes it makes based on the feedback of its own actions [118]. Figure 14 illustrates the necessary elements contributing to this type of learning paradigm. In this context, the agent uses the ML model to take its own action in a specific environment. These actions are interpreted as a reward by a supervisor and a representation of the state, which are returned to it again to accomplish the learning procedures.

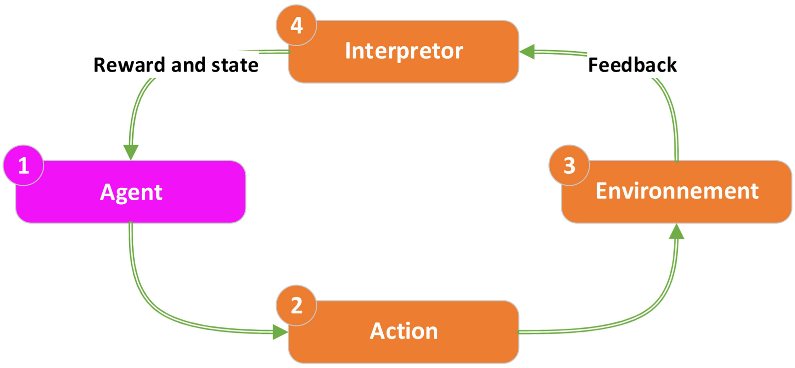


Figure 14. Overview of RL in ML.

RL algorithms are generally categorized into on-policy and off-policy categories, which represent model-based and model-free algorithms. Generally speaking, RL is based on the optimization function $Q(S, a)$ of (9), known as the Bellman equation, which measures the quality of action a and maximizes the reward r that the agent obtains at the state S. π is the probability to find the maximum reward.

$$Q(S_i, a_i) = (1 - \alpha).Q(S_{i+1}, a_{i+1}) + \alpha(r + \gamma.\max(Q(S_{i+2}, a_{i+2}))) \tag{9}$$

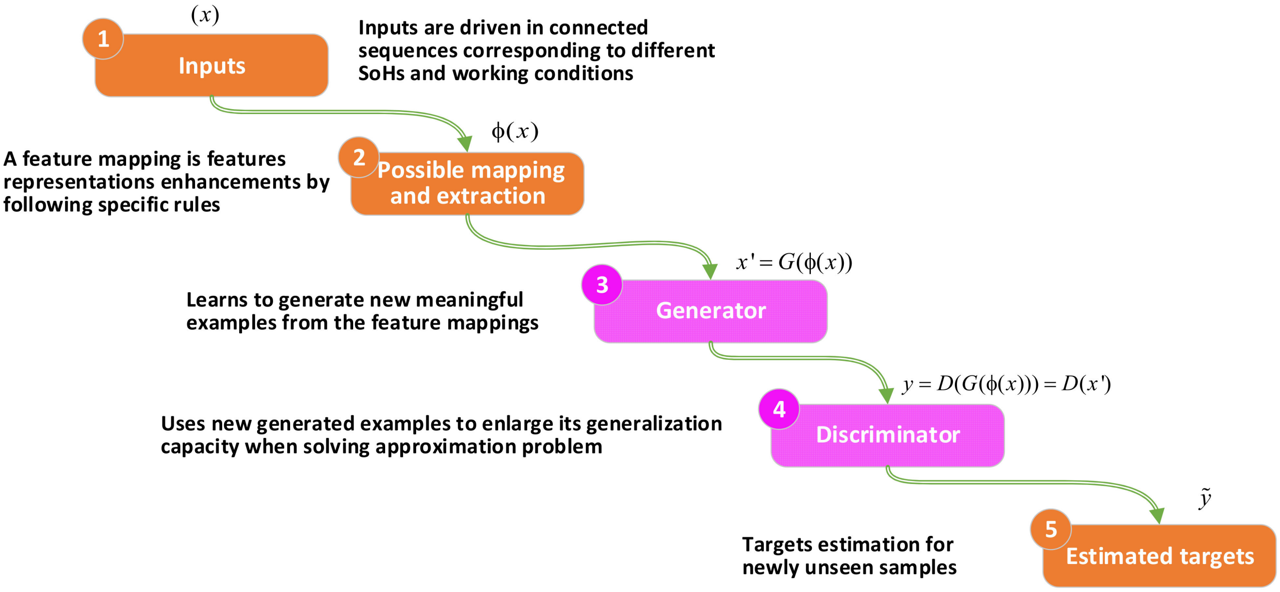
Since the training process is happening at the same instant when data are driven by the actual phenomenon, learning models require being online adaptive ones able to approximate and resist any changes in data to keep generalization preferences. In terms of PHM and specifically for RUL prediction, the purpose of using RL is to be able to make the model self-updatable to learn from its wrong decisions. Practically, in PHM, it is difficult and could be detrimental to learn in real environments. Simulation environments are therefore more suitable in this case. For instance, authors in [119] developed a TL approach that uses an ANN to learn from state, action, and rewards to output the optimal rewards policy. The algorithms target a sequential RUL prediction process of a specific type of pumping system. In [120], authors proposed to study an RL approach using a simulation model of a DC motor and shaft wear. Data are generated from an analytical model by mimicking the real system. The application is supposed to be an SoH assessment when predicting the RUL. In [121], a Bayesian filtering-based deep RL approach was proposed to predict the RUL of aircraft engines.

In PHM, and specifically in real-world applications, RL is meant to be used to make decisions about specific maintenance tasks based on sequential (i.e., just-in-time) learning.

4.5. GMs

GMs are learning machines trained to be able to generate new helpful examples to provide a more meaningful representation than the original feature space. These kinds of models also have the possibility to increase data representation by generating new instances [122]. In RUL prediction, GMs are generally linked to a discriminator for further predictions (Figure 15). A generator could be trained in both supervised and unsupervised ways. The difference between these types of ML modeling and other learning methods such as DL is that the loss function contains in this case two terms related to both generator and discriminator. The loss function of the supervised learning process can be defined as in (10), where G and D are the discriminator and generator functions, respectively. l_x and l_x are loss functions for the generator and discriminator, penalized with discount coefficients δ_x and δ_y, respectively.

$$loss = \delta_x l_x(x, G(x)) + \delta_y l_y(y, \mathrm{D}(x\prime)) \tag{10}$$



(x)
1
Inputs
Inputs are driven in connected sequences corresponding to different SoHs and working conditions
$\phi(x)$
A feature mapping is features representations enhancements by following specific rules
2
Possible mapping and extraction
$x' = G(\phi(x))$
Learns to generate new meaningful examples from the feature mappings
3
Generator
$y = D(G(\phi(x))) = D(x')$
Uses new generated examples to enlarge its generalization capacity when solving approximation problem
4
Discriminator
$\tilde{y}$
Targets estimation for newly unseen samples
5
Estimated targets

Figure 15. Using GMs for supervised learning.

In the PHM case, GANs are thoroughly discussed since they are the most popular ones. Unlike any GM, a GAN has two parts: the generator and the discriminator. The generator is used to generate new samples, different from other GMs such as AEs where the discriminator is used for prediction. GANs discriminator part tries to distinguish fake samples from real ones (i.e., including the main idea behind GANs). The discriminator penalizes the generator loss function for producing false results [123]. GANs were also investigated in PHM. Indeed, in [40], deep GANs were used for data augmentation when predicting the RUL with incomplete data, i.e., the XJTU-SY [124] and the PRONOSTIA datasets, respectively. Authors in [125] used both C-MAPSS and the PRONOSTIA datasets with a deep adversarial approach. In [126], convolutional recurrent GANs were used to assess the RUL of aircraft engines. In [127], a deep recurrent GAN and action discovery were used to estimate bearings RUL using run-to-failure measurements obtained from an accelerated life test. In these contexts, it was proven that GANs are very powerful tools when trying to fulfill the data augmentation condition, specifically during the use of discriminator deciding which data are appropriate for training.

4.6. DA

DA is an area of ML modeling, where the main objective is to train an ML model on a source domain and ensure accurate modeling on a target domain that is substantially different from the original source domain. Among DA methods, TL is the main one used in RUL model reconstruction and prediction. TL is the exploitation of previously trained models or previous knowledge in general about a particular feature space of data in a new learning process. Thus, similarities between previous and current feature space are very important to maximize the generalization of the ML model [128]. There are many types of knowledge transfer including inductive learning, transductive learning, cross-modality, negative learning, and unsupervised TL [21]. In the PHM field, RUL prediction is generally carried out through cross-modality TL. In this case, training weights (i.e., learning parameters in general) are transferred from the previously trained model on similar data, where the most important feature is that prediction of entire loss functions are a combination of both generator and discriminator.

In this context, in [129], an RNN was used to transfer the learning parameters between learning models trained on rich to poor data. Authors' experiments were carried out on a turbofan engine dataset. In [130], authors followed the same methodology to train consensus self-organizing models for predicting the RUL of a turbofan engine. In [37], authors used an LSTM to transfer the learning parameters from different life cycles of the PRONOSTIA dataset in both HI and HS estimation processes. In [131], a transfer component analysis was introduced after well-defined deep features representations with a contractive denoising AE. This transfer mechanism was used to adjust features in the target domain before using least square for SoH assessment with SVM. Experiments of the introduced framework are carried out using the PRONOSTIA dataset. It should be mentioned that TL helps not only in providing generalization from other complete datasets but also in reducing data distribution mismatch between testing and training samples.

Table 4 summarizes all the algorithms discussed in references provided in this section. As previously mentioned, Table 4 also shows that most of the work discussed used the PRONOSTIA and C-MAPSS datasets, which justifies our choice of these datasets as the primary examples in our paper.

Table 4. Summary of the most important used ML tools in RUL prediction.

References	ML Classes	Methods	Datasets/Systems
[67–71]	Conventional ML	SVM	C-MAPSS, Li-ion batteries, PRONOSITA, and IMS
[72–75]		KNN	IGBTs, Li-ion batteries, and other bearings datasets
[39,76–79]		MLP	C-MAPSS, Tool wear, and timing belt in a combustion engine
[1,2,81,82]		ELM	C-MAPSS and an integrated modular avionic system
[27,88,89]	DL	LSTM	C-MAPSS and other bearings datasets
[90–93]		CNN	PRONOSTIA and other bearings datasets
[95–99]		DBN	C-MAPSS, wind turbine gearbox, and supercapacitors
[100–102]		AEs	Li-ion batteries, cutting tool, and other bearings datasets
[104,105]		GNN	C-MAPSS
[119–121]	RL	RL	C-MAPSS, pumping system, DC motor, and shaft wear
[108–111]	ECT and SI	PSO	Li-ion batteries and journal bearing seizure
[113,114,117]		GA	C-MAPSS, Li-ion, IMS, and supercapacitors
[40,125–127]	GMs	GANs	C-MAPSS, PRONOSTIA, and XJTU-SY
[37,128–131]	DA	TL	C-MAPSS and PRONOSTIA

5. Discussion

According to the proposed flowchart highlighting the necessary steps for constructing an RUL prediction model with ML tools (Figure 2) and the above-presented review-based study, important conclusions can be drawn for each class of ML. This section is then devoted to describing these conclusions and the challenges of RUL-based ML model reconstructions.

5.1. Conventional ML

Conventional ML tools are explicitly used to approximate the inputs to the targets. In this context, data are considered ready for training and their representations are satisfactory for building an ML model. However, since the majority of conventional ML models are not dynamic and adaptive ones that are not able to shape data variation during the degradation process, data, in this case, must describe as a noncomplex static process. Otherwise, learning models are likely to fail. In addition, one of the main drawbacks of traditional ML tools is that they require much intervention, especially for data preprocessing to improve its quality (e.g., feature mappings using full rank and kernels mappings) spatially when it does not introduce self-adaptive learning towards data changes. However, in some cases, we may use conventional ML tools to, for example, test the credibility of run-to-failure datasets and their quality, as in the following:

- In terms of ML and concerning the SVM tool, the best way to use them is by following a joint classification–regression scheme. The classification process is dedicated to HS splitting followed by HI prediction. Even for complete data where labels are available, HS splitting before RUL prediction is of great advantage.
- KNN can be used either for HS splitting or direct HI estimation. However, it is wastefully recommended for HS splitting process, especially when data is generally based on unsupervised learning paradigms.
- MLPs and ELM are generally used for direct RUL predictions. Indeed, strengthening such simple algorithms in terms of representation learning and keeping their simplicity is of great advantage in reducing computational costs, as in automatic neural networks with augmented hidden layer (Auto-NAHL) theories [5].

5.2. Advanced DL

DL tools are very useful since they do not require a lot of data preprocessing compared with conventional ML. The main advantage of DL tools is that they are capable of learning from representation, which is useful when extracting necessary patterns. In this context, it can be particularly noted:

- Due to the nature of sequentially driven data, recursive or adaptive learning is required when building ML models. Therefore, we found that LSTM is more popular than CNN when dealing with this type of data. Indeed, LSTM has the ability to control the remembering and forgetting process of both previous and current used training samples with the help of specific gates. Such a feature is not available in CNNs original theories, but it can be added as in convolutional LSTMs or CNNs with recurrent gated units. The most important is that LSTM variants are preferable in such situations due to their powerful capability in beating vanishing gradient phenomena rather than other recurrent networks. It should be mentioned that LTSM could be the best way to determine HI, since it is time-series curve fitting, and prediction problems.
- We cannot deny the accuracy of the CNN capability in feature representations. In fact, this is the reason behind using joint CNN–LSTM. In such a situation, the mapping features of CNNs more effectively contributes to the separation of scattered data that should have similar representations and characteristics. Accordingly, projecting them on SoH evaluation cases, CNNs are more adequate for the HS splitting and classification process.
- DBNs and AEs are recommended for feature extraction, but they do not have the adaptive learning features of LSTMs. This means that when dealing with such algorithms, explainability in terms of real-world applications is then lacking. Therefore, data dynamism must always be considered in such a situation to ensure system prognosability.
- One of the main drawbacks of using deep networks is the computational burden. Besides, learning hyperparameters’ number is huge, making it very difficult to find optimal solutions.

5.3. RL

RL is very important, specifically for online maintenance decision making. However, many drawbacks were reported in the above-discussed literature when building ML models for RUL prediction.

- Most of the studied cases were driven from simulation models (i.e., data already exist), which do not reflect real-world application conditions. In this case, studies are conducted for the purpose of collecting necessary conclusions and not for the real-time recording process investigation.
- RL needs a simulation environment where data comes in sequences and when agent actions are simultaneously executed. It is very difficult to afford this in such a situation due to possible agent errors when learning in a real environment. Subsequently, these errors could lead to catastrophic damages and loss of life.

5.4. ECT and SI

ECTs can be used either for training a prediction model or for hyperparameter optimization. However, according to the above-reviewed works, ECTs are highly recommended for hyperparameters tuning when their number is generally lower than the training parameter one. Besides, ECTs are applicable in ML modeling (i.e., RUL more specifically) and in fast optimizers only if the hyperparameters' number is not large. One of the main drawbacks of ECTs and SI methods is the inability to control the fitness function divergence when the learning model hyperparameters become massive. As previously stated, this massiveness is tightly related to model complexity. The more complex the model is, the more hyperparameters are required. Thus, the more hyperparameters there are, the more difficult the tuning process becomes.

5.5. GMs

Generative models are the best way to improve data distribution and provide more examples to fill the gaps due to the lack of learning patterns, in particular when using run-to-failure measurements obtained from accelerated life tests. However, keeping the balance between loss terms of discriminator and generator is a difficult task. This could lead to the so-called mode collapse that occurs when the generator can only produce a limited type of learning pattern. This could be the result of data quality driving the generator to provide only one type of data [132].

6. Future Improvements and Opportunities

After investigating different classes of ML models, limits, advantages, and disadvantages, this section is devoted to providing important guidelines on improving RUL prediction to move to more realistic conclusions. Accordingly, we targeted many important aspects in terms of real data characteristics (e.g., complexity, availability, and drift) and model characteristics (e.g., complexity, evaluation, and dynamicity) that were previously described in the flowchart of Figure 2.

6.1. Data Characteristics

- Most of the previously discussed datasets in Table 4 are generally obtained through accelerated aging experiments (e.g., bearing and Li-ion batteries), or through simulation models (e.g., C-MAPSS). In this context, the obtained results and constructed model do not appropriately fit the real degradation phenomenon. Therefore, more efforts need to be spent on real-data collection from real-world industrial plants.
- Additional efforts also are needed to provide even more complex, industry-like data for an effective validation when dealing with complex industrial plants. For instance, existing datasets (Table 4) generally do not consider the data heterogeneity phenomena where recorded samples are subject to different constraints of multiple recording rates. Therefore, more effort is needed to address this issue and its impact on the prediction process.
- In real-world applications, data coming from different systems are heterogynous and multisource. Therefore, it is mandatory to provide further data-driven experiments in the context of RUL-based similarity modeling and transfer learning.

6.2. Model Complexity

- Since the RUL prediction process describes a dynamic process that changes over time, offline nonadaptive training algorithms are not appropriate solutions. Therefore, models such as CNN, ELM, and SVM need to consider additive features of dynamic learning, such as GRU for instance.
- Only adaptive algorithms, such as adaptive filters (e.g., least square), LSTM, OSELM, and their variants can be used for RUL prediction.
- GNNs should be further examined to provide insights into their application, specifically where there is an obvious scarcity in their applications.

- For RL, simulation-based virtual reality is helpful to provide more realistic learning rather than traditional off-line simulations.
- For better explaining RUL prediction, it is advantageous to use the early and late prediction evaluation metrics required for conditional maintenance tasks rather than ordinary approximation metrics.

7. Conclusions

In this paper, a systematic guide for predicting RUL was introduced in a review-based study. The main objective of this study was to introduce necessary steps that should be followed to select the best class of ML prediction model for specific run-to-failure data. Different ML models classes were therefore addressed according to the proposed model selection scheme. Besides, necessary steps for constructing the RUL model were also introduced. These steps describe building an ML model starting from data acquisition through processing, training, and evaluation towards RUL prediction. In this review context, important conclusions on the application of ML models on RUL prediction were drawn and limitations were identified. Consequently, important future improvements and opportunities were also discussed. These future prospects were mainly proposed to further improve both data quality and RUL model reconstructions. Data quality prospects targeted data characteristics such as availability, complexity, and drift, while RUL model reconstruction prospects focused on model complexity.

Author Contributions: Conceptualization, T.B. and M.B.; methodology, T.B. and M.B.; software, T.B.; validation, T.B. and M.B.; formal analysis, T.B. and M.B.; investigation, T.B. and M.B.; data curation, T.B. and M.B.; writing—original draft preparation, T.B.; writing—review and editing, T.B. and M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Berghout, T.; Mouss, L.; Kadri, O.; Saïdi, L.; Benbouzid, M. Aircraft Engines Remaining Useful Life Prediction with an Improved Online Sequential Extreme Learning Machine. *Appl. Sci.* **2020**, *10*, 1062. [CrossRef]
2. Berghout, T.; Mouss, L.H.; Kadri, O.; Saïdi, L.; Benbouzid, M. Aircraft engines Remaining Useful Life prediction with an adaptive denoising online sequential Extreme Learning Machine. *Eng. Appl. Artif. Intell.* **2020**, *96*, 103936. [CrossRef]
3. Hu, Y.; Liu, S.; Lu, H.; Zhang, H. Remaining Useful Life Model and Assessment of Mechanical Products: A Brief Review and a Note on the State Space Model Method. *Chin. J. Mech. Eng.* **2019**, *32*, 15. [CrossRef]
4. Liao, L.; Kottig, F. Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction. *IEEE Trans. Reliab.* **2014**, *63*, 191–207. [CrossRef]
5. Berghout, T.; Benbouzid, M.; Muyeen, S.M.; Bentrcia, T.; Mouss, L.H. Auto-NAHL: A Neural Network Approach for Condition-Based Maintenance of Complex Industrial Systems. *IEEE Access* **2021**, *9*, 152829–152840. [CrossRef]
6. Ding, D.; Han, Q.L.; Wang, Z.; Ge, X. A Survey on Model-Based Distributed Control and Filtering for Industrial Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2483–2499. [CrossRef]
7. Kang, S.; Jin, R.; Deng, X.; Kenett, R.S. Challenges of modeling and analysis in cybermanufacturing: A review from a machine learning and computation perspective. *J. Intell. Manuf.* **2021**, 1–14. [CrossRef]
8. Xu, Y.; Sun, Y.; Wan, J.; Liu, X.; Song, Z. Industrial Big Data for Fault Diagnosis: Taxonomy, Review, and Applications. *IEEE Access* **2017**, *5*, 17368–17380. [CrossRef]
9. Hamadache, M.; Jung, J.H.; Park, J.; Youn, B.D. A comprehensive review of artificial intelligence-based approaches for rolling element bearing PHM: Shallow and deep learning. *JMST Adv.* **2019**, *1*, 125–151. [CrossRef]
10. Shang, C.; You, F. Data Analytics and Machine Learning for Smart Process Manufacturing: Recent Advances and Perspectives in the Big Data Era. *Engineering* **2019**, *5*, 1010–1016. [CrossRef]
11. Javed, K.; Gouriveau, R.; Zerhouni, N. State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. *Mech. Syst. Signal Process.* **2017**, *94*, 214–236. [CrossRef]
12. Lipu, M.S.H.; Hannan, M.A.; Hussain, A.; Hoque, M.M.; Ker, P.J.; Saad, M.H.M.; Ayob, A. A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations. *J. Clean. Prod.* **2018**, *205*, 115–133. [CrossRef]
13. Khan, S.; Yairi, T. A review on the application of deep learning in system health management. *Mech. Syst. Signal Process.* **2018**, *107*, 241–265. [CrossRef]

14. Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech. Syst. Signal Process.* **2018**, *104*, 799–834. [CrossRef]
15. Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; Gao, R.X. Deep learning and its applications to machine health monitoring. *Mech. Syst. Signal Process.* **2019**, *115*, 213–237. [CrossRef]
16. Li, Y.; Liu, K.; Foley, A.M.; Zülke, A.; Berecibar, M.; Nanini-Maury, E.; Van Mierlo, J.; Hoster, H.E. Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review. *Renew. Sustain. Energy Rev.* **2019**, *113*, 109254. [CrossRef]
17. Zhang, S.; Zhang, S.; Wang, B.; Habetler, T.G. Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review. *IEEE Access* **2020**, *8*, 29857–29881. [CrossRef]
18. Srikanth, I.; Arockiasamy, M. Deterioration models for prediction of remaining useful life of timber and concrete bridges: A review. *J. Traffic Transp. Eng.* **2020**, *7*, 152–173. [CrossRef]
19. He, B.; Liu, L.; Zhang, D. Digital twin-driven remaining useful life prediction for gear performance degradation: A review. *J. Comput. Inf. Sci. Eng.* **2021**, *21*, 030801. [CrossRef]
20. Wang, S.; Jin, S.; Bai, D.; Fan, Y.; Shi, H.; Fernandez, C. A critical review of improved deep learning methods for the remaining useful life prediction of lithium-ion batteries. *Energy Rep.* **2021**, *7*, 5562–5574. [CrossRef]
21. Niu, S.; Liu, Y.; Wang, J.; Song, H. A Decade Survey of Transfer Learning (2010–2020). *IEEE Trans. Artif. Intell.* **2020**, *1*, 151–166. [CrossRef]
22. Berghout, T.; Benbouzid, M.; Bentrcia, T.; Ma, X.; Djurović, S.; Mouss, L.H. Machine Learning-Based Condition Monitoring for PV Systems: State of the Art and Future Prospects. *Energies* **2021**, *14*, 6316. [CrossRef]
23. Sikorska, J.Z.; Hodkiewicz, M.; Ma, L. Prognostic modelling options for remaining useful life estimation by industry. *Mech. Syst. Signal Process.* **2011**, *25*, 1803–1836. [CrossRef]
24. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, IEEE, Denver, CO, USA, 6–9 October 2008; pp. 1–9.
25. Nectoux, P.; Gouriveau, R.; Medjaher, K.; Ramasso, E.; Chebel-Morello, B.; Zerhouni, N.; Varnier, C. PRONOSTIA: An experimental platform for bearings accelerated degradation tests. In Proceedings of the IEEE International Conference on Prognostics and Health Management, PHM'12, Denver, CO, USA, 18–21 June 2012; pp. 1–8.
26. Zheng, S.; Ristovski, K.; Farahat, A.; Gupta, C. Long Short-Term Memory Network for Remaining Useful Life estimation. In Proceedings of the 2017 IEEE international conference on prognostics and health management (ICPHM), Dallas, TX, USA, 19–21 June 2017; pp. 88–95. [CrossRef]
27. Wu, Y.; Yuan, M.; Dong, S.; Lin, L.; Liu, Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing* **2018**, *275*, 167–179. [CrossRef]
28. Wu, Z.; Yu, S.; Zhu, X.; Ji, Y.; Pecht, M. A Weighted Deep Domain Adaptation Method for Industrial Fault Prognostics According to Prior Distribution of Complex Working Conditions. *IEEE Access* **2019**, *7*, 139802–139814. [CrossRef]
29. Miao, H.; Li, B.; Sun, C.; Liu, J. Joint Learning of Degradation Assessment and RUL Prediction for Aeroengines via Dual-Task Deep LSTM Networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5023–5032. [CrossRef]
30. Nejabatkhah, F.; Li, Y.W.; Liang, H.; Ahrabi, R.R. Cyber-security of smart microgrids: A survey. *Energies* **2021**, *14*, 27. [CrossRef]
31. Xiang, S.; Qin, Y.; Luo, J.; Pu, H.; Tang, B. Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. *Reliab. Eng. Syst. Saf.* **2021**, *216*, 107927. [CrossRef]
32. Li, X.; Ding, Q.; Sun, J.Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab. Eng. Syst. Saf.* **2018**, *172*, 1–11. [CrossRef]
33. Li, H.; Zhao, W.; Zhang, Y.; Zio, E. Remaining useful life prediction using multi-scale deep convolutional neural network. *Appl. Soft Comput. J.* **2020**, *89*, 106113. [CrossRef]
34. Berghout, T.; Mouss, L.H.; Kadri, O.; Hadjidj, N. Regularized Length Changeable Extreme Learning Machine with Incremental Learning Enhancements for Remaining Useful Life Prediction of Aircraft Engines. In Proceedings of the CCSSP 2020—1st International Conference on Communications, Control Systems and Signal Processing, EL Oued, Algeria, 16–17 May 2020; pp. 358–363.
35. Li, X.; Jiang, H.; Liu, Y.; Wang, T.; Li, Z. An integrated deep multiscale feature fusion network for aeroengine remaining useful life prediction with multisensor data. *Knowl. Based Syst.* **2021**, *235*, 107652. [CrossRef]
36. Heimes, F.O. Recurrent neural networks for remaining useful life estimation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 6–9 October 2008; pp. 1–6. [CrossRef]
37. Berghout, T.; Mouss, L.H.; Bentrcia, T.; Benbouzid, M. A Semi-supervised Deep Transfer Learning Approach for Rolling-Element Bearing Remaining Useful Life Prediction. *IEEE Trans. Energy Convers.* **2021**, 1. [CrossRef]
38. Zeng, F.; Li, Y.; Jiang, Y.; Song, G. An online transfer learning-based remaining useful life prediction method of ball bearings. *Meas. J. Int. Meas. Confed.* **2021**, *176*, 109201. [CrossRef]
39. Zhu, J.; Chen, N.; Shen, C. A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions. *Mech. Syst. Signal Process.* **2020**, *139*, 106602. [CrossRef]
40. Li, X.; Zhang, W.; Ma, H.; Luo, Z.; Li, X. Data alignments in machinery remaining useful life prediction using deep adversarial neural networks. *Knowl. Based Syst.* **2020**, *197*, 105843. [CrossRef]

41. Aydin, O.; Guldamlasioglu, S. Using LSTM networks to predict engine condition on large scale data processing framework. In Proceedings of the 2017 4th International Conference on Electrical and Electronic Engineering (ICEEE), Dubai, United Arab Emirates, 8–10 April 2017; pp. 281–285.
42. Guo, L.; Li, N.; Jia, F.; Lei, Y.; Lin, J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing* **2017**, *240*, 98–109. [CrossRef]
43. Xia, M.; Li, T.; Shu, T.; Wan, J.; de Silva, C.W.; Wang, Z. A Two-Stage Approach for the Remaining Useful Life Prediction of Bearings Using Deep Neural Networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3703–3711. [CrossRef]
44. Peng, Y.; Wang, Y.; Zi, Y. Switching State-Space Degradation Model with Recursive Filter/Smoother for Prognostics of Remaining Useful Life. *IEEE Trans. Ind. Inform.* **2019**, *15*, 822–832. [CrossRef]
45. Wen, J.; Gao, H.; Zhang, J. Bearing Remaining Useful Life Prediction Based on a Nonlinear Wiener Process Model. *Shock Vib.* **2018**, *2018*, 4068431. [CrossRef]
46. Klausen, A.; Van Khang, H.; Robbersmyr, K.G. Novel Threshold Calculations for Remaining Useful Lifetime Estimation of Rolling Element Bearings. In Proceedings of the 2018 XIII International Conference on Electrical Machines (ICEM), Alexandroupoli, Greece, 3–6 September 2018; pp. 1912–1918.
47. Chaitanya, B.K.; Yadav, A.; Pazoki, M.; Abdelaziz, A.Y. A comprehensive review of islanding detection methods. In *Uncertainties in Modern Power Systems*; Elsevier: Amsterdam, The Netherlands, 2021; pp. 211–256.
48. Liu, C.; Zhang, L.; Niu, J.; Yao, R.; Wu, C. Intelligent prognostics of machining tools based on adaptive variational mode decomposition and deep learning method with attention mechanism. *Neurocomputing* **2020**, *417*, 239–254. [CrossRef]
49. Guo, R.; Li, Y.; Zhao, L.; Zhao, J.; Gao, D. Remaining Useful Life Prediction Based on the Bayesian Regularized Radial Basis Function Neural Network for an External Gear Pump. *IEEE Access* **2020**, *8*, 107498–107509. [CrossRef]
50. Xu, L.; Pennacchi, P.; Chatterton, S. A new method for the estimation of bearing health state and remaining useful life based on the moving average cross-correlation of power spectral density. *Mech. Syst. Signal Process.* **2020**, *139*, 106617. [CrossRef]
51. Wu, J.Y.; Wu, M.; Chen, Z.; Li, X.L.; Yan, R. Degradation-Aware Remaining Useful Life Prediction with LSTM Autoencoder. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3511810. [CrossRef]
52. Cheng, C.; Ma, G.; Zhang, Y.; Sun, M.; Teng, F.; Ding, H.; Yuan, Y. A Deep Learning-Based Remaining Useful Life Prediction Approach for Bearings. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1243–1254. [CrossRef]
53. Ding, H.; Yang, L.; Cheng, Z.; Yang, Z. A remaining useful life prediction method for bearing based on deep neural networks. *Measurement* **2021**, *172*, 108878. [CrossRef]
54. Loutas, T.; Eleftheroglou, N.; Georgoulas, G.; Loukopoulos, P.; Mba, D.; Bennett, I. Valve Failure Prognostics in Reciprocating Compressors Utilizing Temperature Measurements, PCA-Based Data Fusion, and Probabilistic Algorithms. *IEEE Trans. Ind. Electron.* **2020**, *67*, 5022–5029. [CrossRef]
55. Wang, H.; Ni, G.; Chen, J.; Qu, J. Research on rolling bearing state health monitoring and life prediction based on PCA and Internet of things with multi-sensor. *Measurement* **2020**, *157*, 107657. [CrossRef]
56. Knoebel, C.; Strommenger, D.; Reuter, J.; Guehmann, C. Health Index Generation Based on Compressed Sensing and Logistic Regression for Remaining Useful Life Prediction. In Proceedings of the Annual Conference of the PHM Society, Scottsdale, AZ, USA, 21–26 September 2019; Volume 11. [CrossRef]
57. Wu, B.; Gao, Y.; Feng, S.; Chanwimalueang, T. Sparse Optimistic Based on Lasso-LSQR and Minimum Entropy De-Convolution with FARIMA for the Remaining Useful Life Prediction of Machinery. *Entropy* **2018**, *20*, 747. [CrossRef] [PubMed]
58. Zhang, C.; Lim, P.; Qin, A.K.; Tan, K.C. Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics. *IEEE Trans. Neural Networks Learn. Syst.* **2017**, *28*, 2306–2318. [CrossRef] [PubMed]
59. Duan, Y.; Li, H.; He, M.; Zhao, D. A BiGRU Autoencoder Remaining Useful Life Prediction Scheme with Attention Mechanism and Skip Connection. *IEEE Sens. J.* **2021**, *21*, 10905–10914. [CrossRef]
60. Xu, Y.; Goodacre, R. On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *J. Anal. Test.* **2018**, *2*, 249–262. [CrossRef] [PubMed]
61. Probst, P.; Bischl, B.; Boulesteix, A.L. Tunability: Importance of Hyperparameters of Machine Learning Algorithms. *arXiv* **2018**, arXiv:1802.09596.
62. Feurer, M.; Hutter, F. Hyperparameter Optimization. In *Automated Machine Learning*; Springer: New York, NY, USA, 2019; pp. 3–33. [CrossRef]
63. Singh, J.; Darpe, A.K.; Singh, S.P. Bearing remaining useful life estimation using an adaptive data-driven model based on health state change point identification and K -means clustering. *Meas. Sci. Technol.* **2020**, *31*, 085601. [CrossRef]
64. Berghout, T.; Benbouzid, M.; Mouss, L.H. Leveraging label information in a knowledge-driven approach for rolling-element bearings remaining useful life prediction. *Energies* **2021**, *14*, 2163. [CrossRef]
65. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
66. Evgeniou, T.; Pontil, M. *Support Vector Machines: Theory and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2001; pp. 249–257.
67. Chen, Z.; Cao, S.; Mao, Z. Remaining useful life estimation of aircraft engines using a modified similarity and supporting vector machine (SVM) approach. *Energies* **2018**, *11*, 28. [CrossRef]

68. Ali, M.U.; Zafar, A.; Nengroo, S.H.; Hussain, S.; Park, G.S.; Kim, H.J. Online remaining useful life prediction for lithium-ion batteries using partial discharge data features. *Energies* **2019**, *12*, 4366. [CrossRef]
69. Ordóñez, C.; Sánchez Lasheras, F.; Roca-Pardiñas, J.; Juez, F.J. de C. A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines. *J. Comput. Appl. Math.* **2019**, *346*, 184–191. [CrossRef]
70. Yan, M.; Wang, X.; Wang, B.; Chang, M.; Muhammad, I. Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model. *ISA Trans.* **2020**, *98*, 471–482. [CrossRef]
71. Wu, J.Y.; Wu, M.; Chen, Z.; Li, X.; Yan, R. A joint classification-regression method for multi-stage remaining useful life prediction. *J. Manuf. Syst.* **2021**, *58*, 109–119. [CrossRef]
72. Peterson, L. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
73. Liu, Z.; Mei, W.; Zeng, X.; Yang, C.; Zhou, X. Remaining Useful Life Estimation of Insulated Gate Biploar Transistors (IGBTs) Based on a Novel Volterra k-Nearest Neighbor Optimally Pruned Extreme Learning Machine (VKOPP) Model Using Degradation Data. *Sensors* **2017**, *17*, 2524. [CrossRef] [PubMed]
74. Zhou, Y.; Huang, M.; Pecht, M. Remaining useful life estimation of lithium-ion cells based on k-nearest neighbor regression with differential evolution optimization. *J. Clean. Prod.* **2020**, *249*, 119409. [CrossRef]
75. Motahari-Nezhad, M.; Jafari, S.M. Bearing remaining useful life prediction under starved lubricating condition using time domain acoustic emission signal processing. *Expert Syst. Appl.* **2021**, *168*, 114391. [CrossRef]
76. Murtagh, F. Multilayer perceptrons for classification and regression. *Neurocomputing* **1991**, *2*, 183–197. [CrossRef]
77. Li, W.; Liu, T. Time varying and condition adaptive hidden Markov model for tool wear state estimation and remaining useful life prediction in micro-milling. *Mech. Syst. Signal Process.* **2019**, *131*, 689–702. [CrossRef]
78. Laredo, D.; Chen, Z.; Schütze, O.; Sun, J.Q. A neural network-evolutionary computational framework for remaining useful life estimation of mechanical systems. *Neural Netw.* **2019**, *116*, 178–187. [CrossRef]
79. Khazaee, M.; Banakar, A.; Ghobadian, B.; Mirsalim, M.A.; Minaei, S. Remaining useful life (RUL) prediction of internal combustion engine timing belt based on vibration signals and artificial neural network. *Neural Comput. Appl.* **2021**, *33*, 7785–7801. [CrossRef]
80. Huang, G.B. An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels. *Cognit. Comput.* **2014**, *6*, 376–390. [CrossRef]
81. Gao, Z.; Ma, C.; Zhang, J.; Xu, W. Enhanced Online Sequential Parallel Extreme Learning Machine and its Application in Remaining Useful Life Prediction of Integrated Modular Avionics. *IEEE Access* **2019**, *7*, 183479–183488. [CrossRef]
82. Zhang, B.; Li, Y.; Bai, Y.; Cao, Y. Aeroengines Remaining Useful Life Prediction Based on Improved C-Loss ELM. *IEEE Access* **2020**, *8*, 49752–49764. [CrossRef]
83. Theodoropoulos, P.; Spandonidis, C.C.; Giannopoulos, F.; Fassois, S. A Deep Learning-Based Fault Detection Model for Optimization of Shipping Operations and Enhancement of Maritime Safety. *Sensors* **2021**, *21*, 5658. [CrossRef]
84. Christos, S.C.; Panagiotis, T.; Christos, G. Combined multi-layered big data and responsible AI techniques for enhanced decision support in Shipping. In Proceedings of the 2020 International Conference on Decision Aid Sciences and Application (DASA), Sakheer, Bahrain, 8–9 November 2020; pp. 669–673.
85. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
86. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
87. Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
88. Park, K.; Choi, Y.; Choi, W.J.; Ryu, H.Y.; Kim, H. LSTM-Based Battery Remaining Useful Life Prediction with Multi-Channel Charging Profiles. *IEEE Access* **2020**, *8*, 20786–20798. [CrossRef]
89. Ma, M.; Mao, Z. Deep-Convolution-Based LSTM Network for Remaining Useful Life Prediction. *IEEE Trans. Ind. Inform.* **2021**, *17*, 1658–1667. [CrossRef]
90. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef] [PubMed]
91. Zhu, J.; Chen, N.; Peng, W. Estimation of Bearing Remaining Useful Life Based on Multiscale Convolutional Neural Network. *IEEE Trans. Ind. Electron.* **2019**, *66*, 3208–3216. [CrossRef]
92. Yang, B.; Liu, R.; Zio, E. Remaining useful life prediction based on a double-convolutional neural network architecture. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9521–9530. [CrossRef]
93. Wang, B.; Lei, Y.; Yan, T.; Li, N.; Guo, L. Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery. *Neurocomputing* **2020**, *379*, 117–129. [CrossRef]
94. Webb, G.I.; Fürnkranz, J.; Fürnkranz, J.; Fürnkranz, J.; Hinton, G.; Sammut, C.; Sander, J.; Vlachos, M.; Teh, Y.W.; Yang, Y.; et al. Deep Belief Nets. In *Encyclopedia of Machine Learning*; Springer: Boston, MA, USA, 2011; pp. 267–269.
95. Deutsch, J.; He, D. Using Deep Learning-Based Approach to Predict Remaining Useful Life of Rotating Components. *IEEE Trans. Syst. Man, Cybern. Syst.* **2018**, *48*, 11–20. [CrossRef]
96. Hinton, G.E. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Comput.* **2002**, *14*, 1771–1800. [CrossRef] [PubMed]
97. Peng, K.; Jiao, R.; Dong, J.; Pi, Y. A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter. *Neurocomputing* **2019**, *361*, 19–28. [CrossRef]

98. Pan, Y.; Hong, R.; Chen, J.; Wu, W. A hybrid DBN-SOM-PF-based prognostic approach of remaining useful life for wind turbine gearbox. *Renew. Energy* **2020**, *152*, 138–154. [CrossRef]
99. Haris, M.; Hasan, M.N.; Qin, S. Early and robust remaining useful life prediction of supercapacitors using BOHB optimized Deep Belief Network. *Appl. Energy* **2021**, *286*, 116541. [CrossRef]
100. Bank, D.; Koenigstein, N.; Giryes, R. Autoencoders. *arXiv* **2020**, arXiv:2003.05991.
101. Ren, L.; Sun, Y.; Cui, J.; Zhang, L. Bearing remaining useful life prediction based on deep autoencoder and deep neural networks. *J. Manuf. Syst.* **2018**, *48*, 71–77. [CrossRef]
102. Jiao, R.; Peng, K.; Dong, J. Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Conditional Variational Autoencoders-Particle Filter. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 8831–8843. [CrossRef]
103. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [CrossRef]
104. Li, J.; Li, X.; He, D. A Directed Acyclic Graph Network Combined With CNN and LSTM for Remaining Useful Life Prediction. *IEEE Access* **2019**, *7*, 75464–75475. [CrossRef]
105. Wang, M.; Li, Y.; Zhang, Y.; Jia, L. Spatio-temporal graph convolutional neural network for remaining useful life estimation of aircraft engines. *Aerosp. Syst.* **2021**, *4*, 29–36. [CrossRef]
106. Sloss, A.N.; Gustafson, S. 2019 Evolutionary Algorithms Review. *Genet. Program. Theory Pract. XVII* **2020**, 307–344.
107. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57. [CrossRef]
108. Yu, J.; Mo, B.; Tang, D.; Liu, H.; Wan, J. Remaining useful life prediction for lithium-ion batteries using a quantum particle swarm optimization-based particle filter. *Qual. Eng.* **2017**, *29*, 536–546. [CrossRef]
109. Wang, Y.; Ni, Y.; Li, N.; Lu, S.; Zhang, S.; Feng, Z.; Wang, J. A method based on improved ant lion optimization and support vector regression for remaining useful life estimation of lithium-ion batteries. *Energy Sci. Eng.* **2019**, *7*, 2797–2813. [CrossRef]
110. Ding, N.; Li, H.; Yin, Z.; Zhong, N.; Zhang, L. Journal bearing seizure degradation assessment and remaining useful life prediction based on long short-term memory neural network. *Meas. J. Int. Meas. Confed.* **2020**, *166*, 108215. [CrossRef]
111. Long, B.; Gao, X.; Li, P.; Liu, Z. Multi-Parameter Optimization Method for Remaining Useful Life Prediction of Lithium-Ion Batteries. *IEEE Access* **2020**, *8*, 142557–142570. [CrossRef]
112. Reeves, C.R. Genetic Algorithms. In *Introduction to Genetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 109–139.
113. Listou Ellefsen, A.; Bjørlykhaug, E.; Æsøy, V.; Ushakov, S.; Zhang, H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliab. Eng. Syst. Saf.* **2019**, *183*, 240–251. [CrossRef]
114. Qiu, G.; Gu, Y.; Chen, J. Selective health indicator for bearings ensemble remaining useful life prediction with genetic algorithm and Weibull proportional hazards model. *Meas. J. Int. Meas. Confed.* **2020**, *150*, 107097. [CrossRef]
115. Qiu, H.; Lee, J.; Lin, J.; Yu, G. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *J. Sound Vib.* **2006**, *289*, 1066–1090. [CrossRef]
116. Zhou, Y.; Wang, Y.; Wang, K.; Kang, L.; Peng, F.; Wang, L.; Pang, J. Hybrid genetic algorithm method for efficient and robust evaluation of remaining useful life of supercapacitors. *Appl. Energy* **2020**, *260*, 114169. [CrossRef]
117. Xue, Z.; Zhang, Y.; Cheng, C.; Ma, G. Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression. *Neurocomputing* **2020**, *376*, 95–102. [CrossRef]
118. Sutton, R.S.; Barto, A.G. *Introduction to Reinforcement Learning*; MIT Press: Cambridge, MA, USA, 1998; Volume 135.
119. Bellani, L.; Compare, M.; Baraldi, P.; Zio, E. Towards Developing a Novel Framework for Practical PHM: A Sequential Decision Problem solved by Reinforcement Learning and Artificial Neural Networks. *Int. J. Progn. Health Manag.* **2019**, *31*, 1–15.
120. Jha, M.S.; Weber, P.; Theilliol, D.; Ponsart, J.C.; Maquin, D. A reinforcement learning approach to health aware control strategy. In Proceedings of the 2019 27th Mediterranean Conference on Control and Automation (MED), Akko, Israel, 1–4 July 2019; pp. 171–176. [CrossRef]
121. Skordilis, E.; Moghaddass, R. A deep reinforcement learning approach for real-time sensor-driven decision making and predictive analytics. *Comput. Ind. Eng.* **2020**, *147*, 106600. [CrossRef]
122. Theis, L.; van den Oord, A.; Bethge, M. A note on the evaluation of generative models. *arXiv* **2015**, arXiv:1511.01844.
123. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Adv. Neural Inf. Processing Syst.* **2014**, *27*. [CrossRef]
124. Wang, B.; Lei, Y.; Li, N.; Li, N. A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings. *IEEE Trans. Reliab.* **2020**, *69*, 401–412. [CrossRef]
125. Verstraete, D.; Droguett, E.; Modarres, M. A deep adversarial approach based on multisensor fusion for remaining useful life prognostics. In Proceedings of the 29th European Safety and Reliability Conference, Hannover, Germany, 22–26 September 2020; pp. 1072–1077. [CrossRef]
126. Zhang, X.; Qin, Y.; Yuen, C.; Jayasinghe, L.; Liu, X. Time-Series Regeneration with Convolutional Recurrent Generative Adversarial Network for Remaining Useful Life Estimation. *IEEE Trans. Ind. Inform.* **2021**, *17*, 6820–6831. [CrossRef]
127. Yu, J.; Guo, Z. Remaining useful life prediction of planet bearings based on conditional deep recurrent generative adversarial network and action discovery. *J. Mech. Sci. Technol.* **2021**, *35*, 21–30. [CrossRef]
128. Steiner, G. Transfer of Learning, Cognitive Psychology of. In *International Encyclopedia of the Social & Behavioral Sciences*; Elsevier: Amsterdam, The Netherlands, 2001; pp. 15845–15851.

129. Zhang, A.; Wang, H.; Li, S.; Cui, Y.; Liu, Z.; Yang, G.; Hu, J. Transfer learning with deep recurrent neural networks for remaining useful life estimation. *Appl. Sci.* **2018**, *8*, 2416. [CrossRef]
130. Fan, Y.; Nowaczyk, S.; Rögnvaldsson, T. Transfer learning for remaining useful life prediction based on consensus self-organizing models. *Reliab. Eng. Syst. Saf.* **2020**, *203*, 107098. [CrossRef]
131. Mao, W.; He, J.; Zuo, M.J. Predicting Remaining Useful Life of Rolling Bearings Based on Deep Feature Representation and Transfer Learning. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 1594–1608. [CrossRef]
132. Li, W.; Fan, L.; Wang, Z.; Ma, C.; Cui, X. Tackling mode collapse in multi-generator GANs with orthogonal vectors. *Pattern Recognit.* **2021**, *110*, 107646. [CrossRef]